Tampere University

Atte Sipilä

ANALYSING PRODUCTION FLOW OF DISCRETE MANUFACTURING SYSTEMS USING SIMPLE NODE-BASED DATA

Faculty of Engineering and
Natural Sciences
Master of Science Thesis
March 2019

ABSTRACT

Atte Sipilä: Analysing Production Flow of Discrete Manufacturing Systems Using Simple Node-based Data
Master of Science Thesis
Tampere University
Master's Degree Programme in Automation Technology
March 2019

Optimizing, developing and improving production process of manufacturing companies needs lot of data about the process to support the decision making. The data can be collected with monitoring systems. This thesis aims to justify the need for a new monitoring system which monitors material and production item flow in the manufacturing process using simple formatted data collected from the detection nodes. To justify the need, market situation is explored, and two possible competitors are presented. Based on evaluation of these, there is room for new simple commissioned monitoring system which can be used side by side with possible control system.

Theories ja practices behind key performance indicators and traceability are discussed to get deeper understating of key performance indicator development and flow monitoring. Using the standard for manufacturing key performance indicators, indicators for flow monitoring are presented. The challenge is that the collected data must be kept as simple as possible.

Visualization for key performance indicators are discussed and developed. Choosing the visualization platform happens by comparing ready business intelligence tool and own implementation. The data is generated by simulator which implements the designed interfaces. Analyzing the visualization making process, styles of visualization and costs of different solutions, using the own implementation is selected.

The traceability data is visualized with directed graphs. The examples of the possible visualizations are provided. The realistic looking data is generated with simulator which is using the designed interface to the data collecting module. Visualization challenges of flow of multiple items is solved by making the edges of directed graphs thicker if there are more transactions between nodes.

The decisions are made keeping mind that the flow monitoring system is added to Inspector application of InSolution. The requirements like scalability, commission easiness and clearness and informativeness of visualizations are coming from InSolution. The target is to switch the view point of Inspector monitoring from production devices to production items.

Keywords: KPI, traceability, monitoring, visualization, discrete manufacturing

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

TIIVISTELMÄ

Atte Sipilä: Tuotantovirtauksen analysointi yksinkertaisella tuotannon solmupisteisiin perustuvalla datalla kappaletavara-automaatiossa
Diplomityö
Tampereen yliopisto
Automaatiotekniikan diplomi-insinöörin tutkinto-ohjelma
Maaliskuu 2019

Tuotantojärjestelmien optimointi ja kehitys valmistavassa teollisuudessa vaatii paljon informaatiota tuotantoprosessista tukemaan päätöksen tekoa. Informaatiota voidaan kerätä tuotannon seurantajärjestelmien avulla. Tämän työn keskeisenä tavoitteena on perustella tarve uudelle seurantajärjestelmälle, joka seuraa tuotannon virtausta käyttäen yksinkertaista, tuotannon solmukohdissa kerättyä dataa. Tästä syystä työssä tutkitaan markkinoilla olevia kilpailevia järjestelmiä ja esitellään näistä kaksi. Arvioimalla kaupallisia järjestelmiä tehtiin päätös, että markkinoilla on tilaa uudelle helposti käyttöönotettavalle ja ohjausjärjestelmän rinnalla käytettävälle seurantajärjestelmälle.

Suorituskykyindikaattorien (KPI) ja jäljitettävyyden teorioita ja käytäntöjä tutkitaan, jotta saadaan syvempi ymmärrys indikaattorien kehittämisestä ja tuotantovirtauksen seurannasta. Teollisuuden suorituskykyindikaattoreille kehitettyä standardia käytetään apuna indikaattorien suunnittelussa ja esittelyssä. Tarve kerätä mahdollisimman yksinkertaista informaatiota aiheuttaa omat haasteensa suorituskykyindikaattorien kehittämiseen.

Työssä tutustutaan suorituskykyindikaattorien visualisointiin ja suunnitelluille indikaattoreille kehitetään mahdollisia visualisointeja. Visualisointialustan valinta tapahtuu vertailemalla valmista liiketoimintatiedon hallintaan suunniteltua järjestelmää ja omaa toteutusta. Käytettävä informaatio luodaan simulaattorilla, joka toteuttaa työtä varten suunnitellun rajapinnan tiedonhallintamoduuliin. Kun verrataan visualisointiin tarvittavaa ja käytettävää työtä, visualisoinnin tyylejä sekä toteutuksen kustannuksia, ei valmiilla ja kalliilla järjestelmällä saavuteta tarvittavia etuja. Tästä syystä visualisoinnit päätetään toteuttaa itse.

Kappaleen virtausinformaatiota visualisoidaan suunnattujen graafien avulla, joista annetaan muutama esimerkki. Realistista virtausinformaatiota generoidaan simulaattorilla, joka toteuttaa todelliseen käyttöön suunnitellun rajapinnan tiedonhallintamoduuliin. Kun käytetään suunnattuja graafeja usean kappaleen virtauksen visualisointiin, syntyy haasteita visualisoinnin selkeyden kanssa. Nämä ratkaistaan käyttämällä suunnatussa graafissa paksumpia vektoreita kuvaamaan suurempaa virtausta kahden solmukohdan välillä.

Työssä tehdyt päätökset on tehty huomioiden, että tuotantovirtauksen seurantajärjestelmä yhdistetään InSolutionin Inspector ohjelmistoon. Työssä huomioitavat vaatimukset, kuten järjestelmän skaalautuvuus, käyttöönoton helppous sekä visualisointien selkeys ja informatiivisuus tulevat InSolutionilta. Työn tarkoituksena on esittää keinoja, joilla Inspectorin koneiden seurantajärjestelmän rinnalle voidaan luoda kappaleiden seurantaan tarkoitettu järjestelmä.

Avainsanat: KPI, jäljitettävyys, monitorointi, visualisointi, kappaletavara-automaatio

PREFACE

This thesis is written in three different continents during three different years. Therefore, I am very happy to finally finish it. Hopefully my interest to automation, process improvement and industrial indicators can be read between the lines.

I want to thank my superior Juha Katajisto from subject, guidance and motivation, and Olli-Petteri Hirvonen from cooperation during interface design and encouraging during the writing process. Thanks for Jose L. Martinez Lastra for comments helping me to structure the thesis.

Finally, special thanks to my beautiful wife Jasmin from support, patience and love.

Tampere, 7.3.2019

Atte Sipilä

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	Automatic Data Collection
ANSI	American National Standards Institute
API	Application Programming Interface
B2MML	Business to Manufacturing Markup Language
BI	Business Intelligence
BOM	Bill of Materials
BOD	Business Object Document
BSC	Balanced Scorecard
CI	Continuous Improvement
CMM	Coordinate Measuring Machine
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CSV	Comma Separated Values
DTD	Document Type Definition
ECR	Efficient Consumer Response
EEC	European Economic Community
EPC	Electronic Product Code
ERD	Entity Relationship Diagram
ERM	Entity Relationship Modeling
EU	European Union
FIFO	First in First Out
FIPS	Federal Information Processing Standard
FMS	Flexible Manufacturing System
GPD	Gross Domestic Product
GUI	Graphical User Interface
HTML	Hypertext Markup Language
I/O	Input/Output
IEC	International Electrotechnical Commission
IDEFIX	Integration Definition for Information Modeling
II-RFID	Intelligent and Integrated RFID
IISS	Integration Information Support System
InfoVis	Information Visualization
IoT	Internet of Things
ISA	International Society of Automation
ISO	International Standardization Organization
ISV	Independent Software Vendors
KPI	Key Performance Indicator
KPI-ML	Key Performance Indicator Markup Language
MES	Manufacturing Execution System
MESA	Manufacturing Enterprise Solutions Association
MIT	Massachusetts Institute of Technology
MOM	Manufacturing Operations Management
MTBF	Mean Time Between Failures
NC	Numerical Control
OAGiS	Open Applications Group Integration Specification
OEE	Overall Equipment Effectiveness
OLE	Object Linking and Embedding

OPC	OLE for Process Control or Openness
OSHA	Occupational Safety and Health Administration
PaaS	Platform as a Service
PDF	Portable Document Format
PI	Performance Indicator
PLC	Programmable Logic Controller
PMS	Performance Measurement System
REST	Representational State Transfer
RFID	Radio Frequency Identification
SaaS	Software as a Service
SCM	Supply Chain Management
SDK	Software Development Kit
SLP	Systematic Layout Planning
SPC	Statistical Process Control
SQL	Structured Query Language
UA	Unified Architecture
UML	Unified Modeling Language
VPN	Virtual Private Network
WEB	World Wide Web
WIP	Work in Progress
XML	Extensible Markup Language

1. INTRODUCTION

This chapter introduces the background of the thesis and justifies work by defining the targets. Research environment is also presented.

1.1 Background and Motivation

Over 28 million people in Europe are working on manufacturing companies. These companies generate about 20 % of the output of EU (Jovane et al. 2009). In 2016, manufacturing companies provides about 20.3 % of Gross Domestic Product (GPD) of Finland (EK 2017). That means that there are a lot of potential in the manufacturing field to tune and monitor existing systems. Decision-makers need lots of data and information about the production system to meet higher customer needs, higher quality requirements and higher production efficiency expectations. One challenge of utilizing available data is to find the most relevant information from the huge amount of gathered data (Rakar & Zorzut 2010). Data can be collected anywhere and detecting the right ways to collect and utilize the data to reach the targets is important and fascinating.

Designing suitable and informal key performance indicators (KPIs), the data can be processed and presented in an effective way. In many cases, a good key performance indicator presents the information as a simple number, which is easy to understand, and which presents the state of machine or production in an understandable way. By visualizing the data, different trends can be seen more easily. Visualizations also offer huge amount information fast without analyzing thousands of lines of data.

I have worked couple of years as subcontractor in a big FMS manufacturer. There I have learned to know a lot about manufacturing industry all over the world. The importance of KPIs and reporting data to ERP systems is grown during the years. Traceability of items seems to be very important, at least in the companies which are manufacturing parts for airplanes or cars. Therefore, the software providers should be able to provide accurate data reliable. That is why I find the area of thesis very interesting and useful. Finding the accurate and right ways of developing important KPIs might be huge competition advantage in the future. Using the standards makes providing the KPIs more sustainable and effective.

1.2 Justification

There are millions of ways to collect, analyze and utilize data, and therefore focusing on specific data and monitoring methods must be made. The ways to monitor flow of production items in the manufacturing system using as simple data as possible is explored in this thesis. Using simple node-based data allows adapting monitoring system for multiple different manufacturing plants and layouts. By analyzing the flow, multiple important production indicators, such as throughput time and WIP storage, become available. Therefore, defining and designing the used KPIs should be made.

1.3 Research Objectives

To monitor, analyze and visualize production flow, four steps can be recognized. At the first step, movements of the items are tracked with sensors and readers. In this thesis, flow is monitored using nodes, which can be devices, buffers or checkpoints with attached readers to detect production items. Then at the step2, the data is sent to the service which stores the data to a database allowing later queries. Historical data analyze is also possible if data is stored. At the third step, data is queried from the database for analyzing it. Wanted KPIs can be designed and implemented at this point. At last, analyzed data is visualized for the user, using graphs and numbers. By making the visualization interactive, more value can be offered to the users when the data can be explored. Typically, different kind of dashboards are used as Graphical User Interface (GUI) for presenting the visualizations. These four simplified steps can be seen from figure 1.

Figure 1. *Simplified process for presenting production flow data*

This thesis is concentrating mostly on the steps three and four. First two steps are introduced later in this thesis, but the design and implementation are part of another thesis. Both theses are implemented individually, but data query interface must be agreed together. Following questions are objectives of this thesis:

- How to design and implement effective key performance indicators?
- Are there any standards for key performance indicators?
- What is traceability in production?
- How to monitor production flow of discrete manufacturing systems using node-based data?
- How to form key performance indicators from simple node-based data?
- Selecting and designing right key performance indicators for flow monitoring system
- How key performance indicators and traceability data can be visualized?
- Are there already equivalent commercial solutions on the market?
- Should visualizations be implemented itself or are there any ready solutions?

1.3.1 Research Environment

This thesis is made for InSolution Oy which is automation company located in Tampere Finland. Over the last 14 years, InSolution has made challenging industrial projects for customers in 52 different countries. InSolution has developed production monitoring system called Inspector which can be added to the system without control system changes. Inspector specializes of analyzing states of production devices.

Currently, Inspector collects data from production devices such as machine tools and robots. The connection can be made almost to any manufacturing device using I/O, OLE for Process Control Unified Architecture (OPC UA) or custom-made connection. The structure of Inspector is modular which enables extendable after first commissioning. The Inspector operates as cloud service.

The main target of Inspector is to provide reliable and accurate information about the production so that customers can take the most out of their manufacturing system and gain profitability without expensive device investments. Using Automatic Data Collection (ADC), Inspector provides data which is always available and always valid in real time.

Inspector helps to detect a wide range of indicators which are reflecting the manufacturing environment. Inspector provides information about bottlenecks of production and reveals the unrealized potential of it. For example, customer can detect availability and utilization of production devices which helps to increase number of production hours. Inspector analyzes the collected data to construct key performance indicators (KPIs) like Overall Equipment Effectiveness (OEE), utilization rate, Mean Time Between Failures (MTBF) and failure counts.

Inspector reports the collected and analyzed data with HTML5-format (Hypertext Markup Language) allowing access to the stored data with any device with a web browser. Data is provided on both production history and the current state of production. Figure 2 is screenshot from Inspector for understanding the current situation.

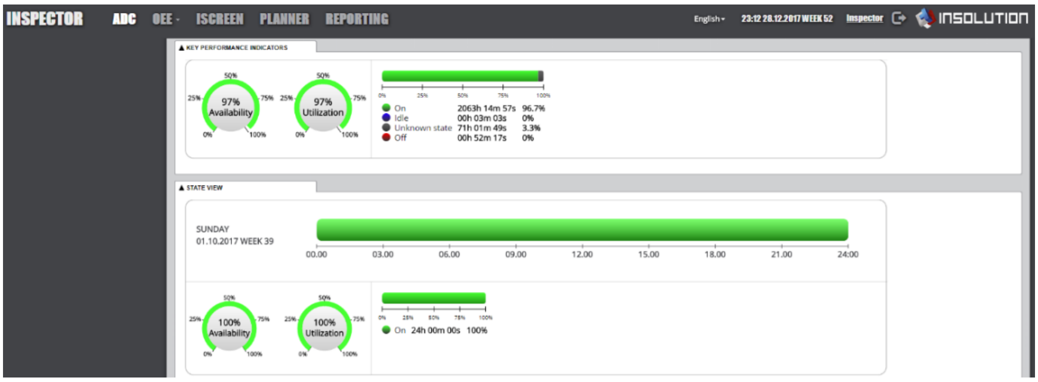


Figure 2. *Screenshot of Inspector ADC view*

The data from the production devices is provided by sensors and Programmable Logic Controllers (PLCs). PLC reads the data from the sensor and sends it to web service. Both wired and wireless communication can be used.

Because Inspector monitors the state of the device, it can detect, when the device is available, when it is running and which NC program it is using. Therefore, the KPI values that Inspector gives are for devices also. This thesis discusses about the flow monitoring and the main target is to monitor the flow of the production items through the manufacturing system and provide accurate information about items and production flows. Also, items with a status of Work in Progress (WIP) can be reported. The motivation of this thesis is to move the observation point of monitoring system from production devices to production items. The preliminary name of flow monitoring system is Inspector Flow Analytics.

Challenge is to find clever ways to exploit the collected flow data to make effective analyzes and to construct important KPIs. Visualization of provided data is very important for usability, and different kind of graphs is used and developed for that. For example, directed graphs are designed to be used for material flow visualization.

The idea is to retain the scalability, ease-of-use and modularity of existing Inspector. To make the Inspector Flow Analytics functional well with original Inspector, it will work on same cloud service and utilize same databases and same equipment. Also, same coding principles are suggested but some changes might be needed. Figure 3 presents the simplified structure of Inspector.

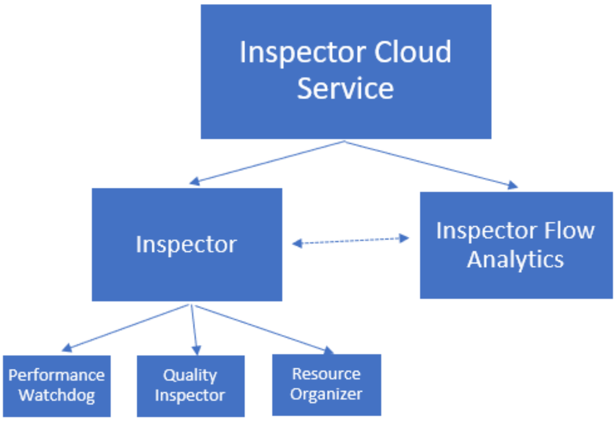


Figure 3. *Inspector Cloud Service structure after adding Inspector Flow Analytics next to original Inspector*

The designed flow monitoring system works under Inspector Cloud Service but is still stand-alone entity allowing installation of only original Inspector or only flow monitoring system. These services could also work together which allows production data exchange between services.

1.4 Thesis structure

Thesis is divided to three parts. First, the literature and best practices review presents KPIs, different standards for them and KPI design methods. Production traceability and production flow monitoring is discussed after that. At the end of literature review, data collection and data visualization methods are presented.

Second part is presenting possible competition on the production flow monitoring area. One of the major parts of this thesis is to decide if there is need for implementing a flow monitoring system or are there already enough competition in the market. Also, possible third-party visualization platforms are presented. Final part of the thesis is focusing on decisions and implementations for flow monitoring system which includes design of KPIs. Also, interfaces and visualizations are demonstrated there.

2. LITERATURE AND INDUSTRIAL BEST PRACTICES REVIEW

This chapter gives overview for key performance indicators and standards related to them. The best practices for KPI design and utilization is also presented. Production item flow in discrete manufacturing is discussed and the term traceability is linked with the production flow monitoring. Practices and methods for collecting, analyzing and visualizing production flow data are also presented.

2.1 Production Improvement with the Help of Key Performance Indicators

This chapter focuses on key performance indicators. ISO 22400 and KPI-ML standards for KPIs are presented to give good understanding of the KPI usage in the manufacturing. For example, ISO 22400 gives helpful tools for defining and presenting KPIs while KPI-ML gives good base for XML-based KPI usage and transaction. The types, history and development of the KPIs are discussed to get better understanding of the importance and distribution of the KPIs in manufacturing.

2.1.1 Overview of Production Measurements

Higher expectations and quality requirements, higher customer needs (Rakar & Zorzut 2010) and bigger competition globally (Jovan & Zorzut 2006; Effendi et al. 2014) are causing pressure to improve the performance of the production. Mulrane arguments in his blog (2016) that improvements can't be done without measurements, or at least improvements are harder to do without up-to-date data about system current performance. Usually improving production performance means that production output or return on investments is rising without big investments. Measurements of current production is important because these can point the weak links of the system (Mulrane 2016) or reveal ineffective ways of doing things. However, the production improvements and analysis internally are not the only way to use measurements but also benchmarking own performance with the similar companies can be utilized (del-Ray-Chamorro et al. 2003).

Performance measurement of a production line or single machine can be done multiple ways. For example, measuring manually with pen and paper is possible, but to get more precise measurements, sensors and software are usually more effective solutions allowing access to data at any time and anywhere (Staniszewski et al. 2014). Also, data collected automatically is more reliable than manually collected and therefore using ADC is suggested. However, data collection and analysis management can be claimed to be an indispensable tool for companies, even if the data collection system is very simple

(Staniszewski et al. 2014). That is one reason why Staniszewski et al. (2014) recommends small companies, which cannot afford ADC for production, to collect data manually. In their example system, operators enter data directly to Excel forms, and even this uncomplicated way has improved the production knowledge of the business managers of the company.

Lukkari (2018) points out a problem where a production manager struggled to follow the production status of the manufacturing – even when the manufacturing was automated. In the example, the production manager needed to discuss with all the operators to know the status of the production orders. Lukkari (2018) presents that with the help of IoT and by monitoring the manufacturing system with the modern monitoring system, the production manager was be able to do the production analyses from the screen without taking a time-consuming walk around the factory. Therefore, with good monitoring system, a lot of time and money can be saved when the status of the production is analyzed.

The measurements itself do not give needed information about production to make improvements effectively. Gathering data is quite fast and there are multiple different methods for it nowadays. Different kind of wireless sensors and networks, radio-frequency indicators, and even smart phones, tablets and laptops are utilized as data gathering tools (Kang et al. 2016). That makes effectively utilization of data a challenge, not at least because the companies are complex and have multiple projects and functions at the same time (Keeple et al. 2003). With process knowledge and with proper data analyze, key performance indicators for the process can be designed and implemented. However, Rakar and Zorkut (2010) remind that KPIs are only one viable way to utilize gathered production data. Nevertheless, with help of right KPIs, decision-makers can make right decisions to help the business to go to the right direction (CA 2015; Mulrane 2016).

ISO 22400 defines key performance indicator as a quantifiable level of achieving a critical objective (ISO 22400:2:2014; Johnsson & Kirsch 2014). Key performance indicators, or at some references only performance indicators (PI), are items of information (Fitz-Gibbon 1990) which are collected to track the performance of the system, person, software or anything under interest. KPI is usually rate, index, percentage or another comparison where an item of information is measured at regular intervals and compared to one or multiple criterions (Jovan & Zorzut 2006). With the help of KPIs, companies can detect their strengths and weaknesses (Effendi et al. 2014). On the other hand, with the KPIs, companies can measure the gap between a current performance and target performance (Weber & Thomas 2005). KPIs can also help companies to achieve certain short-term and middle-term targets like increased motivation of employees, better results in safety of the process and better realization of production planning and scheduling (Rakar et al. 2004). Also, lean manufacturing and waste eliminating can be understood and supported with the help of utilizing KPIs (ISO 22400:2:2014).

It is good to understand that metrics and KPIs are not the same thing. KPI is a metric which gives information about the organization or company performance related to the objectives or targets (Kaushik 2010). For example, if someone wants to measure followers in social media and see how the amount is chancing over a time, collected data forms only a metric. It is not reflecting performance against any target. But if someone is marketing in social media to gain more followers, data about number of followers can form a KPI because then the data is giving information about performance related to the goal.

There are lot of general key performance indicators which can be selected to many companies and processes as such (Rakar et al. 2004; Rakar & Zorzut 2010). Examples of this kind of general KPIs are presented in table 1 which is compiled based on Lindberg et al. (2015) article. Of course, KPIs presented in table 1 aren't fully functional in all processes but these KPIs can be applied to many processes without major modifications.

Table 1. *Examples of general key performance indicators based on Lindberg et al. (2015)*

KPI	Description
Availability	Availability of the machine or the device reveals a time when the machine is in a functional state or in an idle state. Value is usually a percent of the overall time.
Number of alarms over a time period	Number of alarms which occurs in the system or in the single device over a time period. Value is number.
Percentage of full quality products of the production	Full quality products compared to all products produced. Value is a percentage.
Share of time when buffer level is over 95% of buffer size	The share of time over a time period when there are so many items in the buffer that buffer level is over 95% of total buffer size. Value is time unit.

As seen from the table 1, the range of general KPIs is quite wide. If there is not a ready or general indicator from the process, a new specified KPI can be implemented. KPI does not have to be general but it can be related to very specific properties of processes (Rakar et al. 2004; Rakar & Zorzut 2010). That means that KPI does not need to be reusable in other processes. The only thing that matters is that KPI measures something which interests and gives valuable information about the process to support decision making.

Collected data can be used to design multiple key performance indicators instead of only one, which means that collecting data reliable and continuously can be very important also for the future use of it. Rakar et al. (2004) implemented a production information system in their case study. In this study, they were able to use the collected data for several different KPIs. Using the data effectively makes the data collection more reasonable and accountable. Of course, selecting the right data from a mass of production data needs lot of knowledge about the process. It is hard to detect all the points affecting to the production without further understanding of it (Rakar et al. 2004).

Often companies have implemented performance measurement system (PMS) for evaluating performance of manufacturing operations and activities. KPIs are then designed for supporting the strategic goals. Therefore, there are different kind of correlations between KPIs which means that there is mutual relationship between different indicators. KPIs are reflecting one aspect of the manufacturing performance and an individual KPI is not independent. Understanding and utilizing the relationship between KPIs is critical for continuous improvement (CI) of the production system. Investigating and identifying relationships can help for more effective usage of current KPIs or helps the process of defining new, better KPIs. (Kang et al. 2016)

Usually companies define own warning limits and thresholds for KPI values. Then the measuring system or someone analyzing the value can pop up an alarm or warning when the limit or the threshold is reached to inform that improvements to quality or efficiency must be performed. That shows one limitation of KPIs – they are not performing anything alone, but can inform when performance or quality is dropped, and improvements must be done. Analyzing KPI values can also reveal trends of process or equipment before break downs. (ISO 22400:2:2014) This makes KPIs useful tools also for maintenance. For example, measuring vibration of machine tool during specific process can form KPI which is informing mean vibration during processing. If the KPI value is increasing, it might mean that machine tool needs maintenance.

2.1.2 Short Introduction to History

History of KPIs is long and they were first used as early as 3rd century when the performance of the royal family was measured by the emperors of the Chinese Wei Dynasty. At the 1800s, the KPIs were introduced in the industry by Scottish miller. The miller used colorful wood cubes which sides were painted with many colors. The cubes were then placed above workstations of each worker. The more modern KPIs and methods were developed at the 1900s when military and industry needed better performance indications. Not until the beginning of the 1990s, the KPIs were usually reflecting the performance of individuals instead of performance of the company. (Ofori-Boateng 2017)

Introduction of Balanced Scorecards (BSC) in 1990 caused the next big step for KPIs (Ofori-Boateng 2017). Balanced Scorecards are strategic planning and management systems which are, for example, used to present organization strategies, prioritize services, projects, products, and measure KPIs (BSI 2018). BSC is first introduced by Kaplan and Norton in 1996 and it integrates financial KPIs with other KPIs to make four-dimension scorecard (del-Ray-Chamorro et al. 2003). Today, KPIs are spread to everywhere and to every kind of industry (Ofori-Boateng 2017) making the understanding and using the KPIs more important.

2.1.3 Key Performance Indicator Types

It is not easy to sort out KPIs to distinct types or to different use cases because almost everything can be measured and everything which can be measured, can be turned into KPI if some target for the measurement can be identified. Often KPIs are spoken in industrial business but for example, Fitz-Gibbon (1990) develops and implements KPIs for the educational environment. Scoreboard (2018) presents in their website examples of used KPIs in departments and in different industries. Examples of departments are Customer Service Departments and Sales Departments. Departments can use KPIs as well as industries. For industries, twenty US Government's major industry categories are introduced. Table 2 lists two possible KPIs for five of these twenty industries.

Table 2. *Examples of possible KPIs used in 5 different industries adapted from Scoreboard (2018)*

Industry	Key Performance Indicators
Construction Industry	- Number of accidents - Percentage of unapproved change orders
Finance and Insurance Industry	- Accounts payable turnover - Gross profit margin
Manufacturing Industry	- Labor as a percentage of cost - Percentage reduction in defect rates
Professional, Scientific, and Technical Services	- Average percentage of CPU utilization - Mean time between failure (MTBF)
Retail Trade Industry	- Gross profit percentage - Salary overtime percentage

Table 2 gives a good understanding how widely KPIs can be used. KPI can be well-known like gross profit margin or more special and unique like attention rate of online courses in the educational services industry.

Lindberg et al. (2015) handle the types of industrial KPIs from another perspective. They divide the KPIs based on the use of them and shares them to eight different classes. The classes are energy, raw-material, operation, control performance, inventory and buffer utilization, maintenance, planning and equipment KPIs. Table 3 introduces the classes, gives some brief description for each of them and presents possible KPI for every class.

Table 3. *8 different KPI classes based on Lindberg et al. (2015)*

Class	Description	Example KPI
Energy KPIs	Different forms of energy like gas, coal and oil.	Energy output / Energy input
Raw-material KPIs	Raw-materials of the products. Raw-material can also be water, chemicals, etc.	Waste deposit / Produced output
Operation KPIs	Most important operation KPI is OEE (Overall Equipment Effectiveness) and individual parts of it.	OEE
Control performance KPIs	Production quality, speed, equipment wear, etc. may be influenced by control performance.	Number of control loops in manual mode / total number of control loops
Maintenance KPIs	Maintenance affects to production. If there are too little maintenance, lost production occurs because of unplanned stops. With too much maintenance, production lost is caused by maintenance breaks.	Maintenance costs / Produced output over a time period
Planning KPI	Plant capacity utilization is impacted by planning and scheduling.	Integrated sum of only positive values of (planned – actual production) over a time period
Inventory and buffer utilization KPIs	Inventory management is an important part of manufacturing because too large inventories are expensive and too small may cause production disturbances.	Throughput rate / Average Inventory
Equipment KPIs	Following equipment, like machine tools is a common source of KPI.	Equipment wear which can be based on operating hours, speed, load or startups.

Even though only one possible KPI of each class is introduced, the possible amount of KPIs in each class is voluminous. Furthermore, must be noticed that planning KPIs has some challenges because KPI calculations of deriving the optimal production plan and comparing it to the actual production are out of KPI scope. Bonding the plan to the actual production is suggested instead of deriving optimal plan. (Lindberg et al. 2015)

Keeple et al. (2003) divides KPIs based on how easy the data gathering for the KPI is based on the source of the data. If the data needed for KPI is collected outside the organization, the company does not have direct control possibility and the data can be based on estimations. These kind of external KPIs can still be very important and informative for the company. Keeple et al. (2003) identifies three classes of KPIs which are in-house, management, and business partners and product indicators. Figure 4 presents different data collection objects so that internally measured are at the left side of the graph and externally measured at the right side of the graph. Objects from where data is easily collected are at the bottom of the graph and at the top of the graph are objects from where data collection is more complex. Figure 4 is copied from the paper of Keeple et al. (2003).

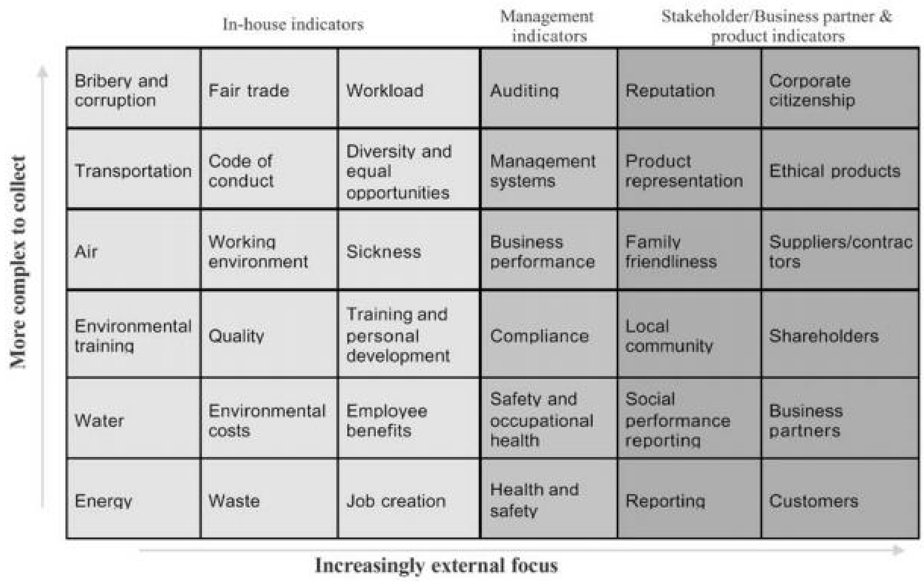


Figure 4. *Different data collection objects for creating KPIs classified based on data collection sources and data collection complexity (Keeple et al. 2003)*

In-house indicators are related to manufacturing and workers and they can be constructed from data which can be acquired inside the company. Even if the data can be gathered near does not mean that collecting is easy - collecting is only possible. For example, collecting data about energy usage is much easier than collecting data from workload of a single worker. Management indicators have connection to internal measurements but also external measurements. The most external focus indicators are business partner and product indicators. Thought the data is coming from outside the company, it does not necessary mean that collecting it is complex. The data about customers of the company or about business partners can sometimes easily be accessible. But some indicators are much more difficult to measure, and the data is based only on surveys or judgments and estimations. For example, the data about company or product reputation is hard to gather because it is complex as a concept.

KPIs can also be separated roughly at the top-level of the industry to plant-level operational KPIs and to business or financial-level KPIs (del-Ray-Chamorro et al. 2003; Fraser 2006). The separation is quite clear because often operational and business KPIs are conflicting. Good example about the conflict between these layers is the outsourcing of production to regions and countries with lower costs. While at business-level KPIs of lower production costs and more value-adding work at headquarter can look good, at operational level KPIs of cargo costs, high inventories and possibility of lower quality are not making outsourcing look very attractive. Sometimes improving business-level KPI can make some other operation level KPI going down. (Fraser 2006) In the whitepaper of CA (2015), the KPIs are separated for four key areas of companies. These areas are service delivery, financial, sales and customer satisfaction. The service delivery can be thought as a production in the manufacturing industry.

This thesis allows separating KPIs based on the data and data gathering methods. Originally, Inspector is providing KPIs based on data gathered from machines and devices, and KPIs are usually device based. Examples of these kinds of KPIs are availability, idle time and OEE of the device. When data is gathered by following the flow of the production items, KPIs based on the production flow can be identified. Examples of these kinds of KPIs are throughput time, output rate and average inventory. While device based KPIs give valuable information about machines and devices, give flow based KPIs information about production and inventory.

2.1.4 Categorization and Relationships

Kang et al. (2016) presents in their research hierarchicalization and categorization of KPIs. The need for categorization raised from the need of detecting the intrinsic relationship between KPIs. The figure 5 represents the three categorized levels of KPIs. Kang et al (2016) also notes that this kind of categorization might not be specific and different kind of relations can be found and developed.

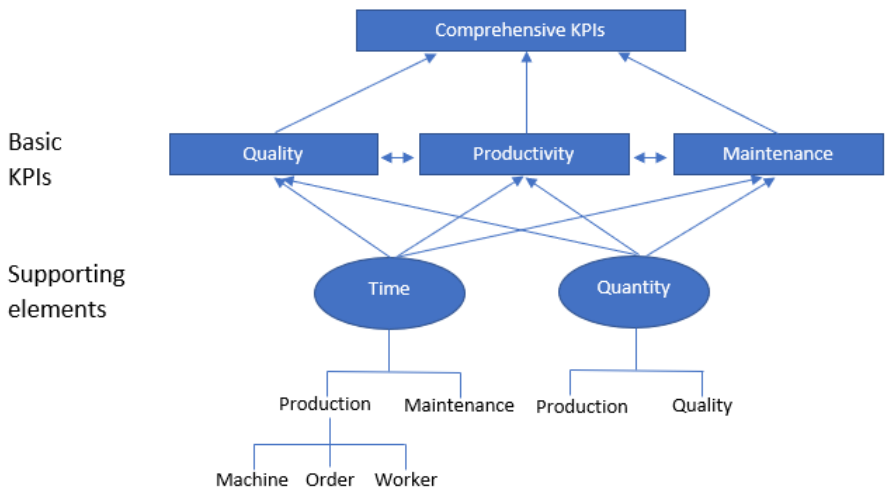


Figure 5. *KPIs categorization based on Kang et al. (2016)*

The first level is supporting elements, which are, for example, direct measurements. Data which is collected directly from the production can be thought as a supporting element. The supporting elements are divided to time and quantity groups. Quantity measurements, like scrap quantity and rework quantity, based on production and quality belongs to the quantity group. The time durations in manufacturing systems can be measured based on maintenance and production, and these measurements belongs to the time group. The time measurements can be based on machines, production orders or workers. Examples of time elements are planned busy time, planned unit setup time and actual unit idle time. (Kang et al. 2016)

The basic KPIs are divided to three groups which are quality, productivity and maintenance. Kang et al. noticed that this grouping is not the only valid option and for example standard ISO 22400:2 uses different grouping. These KPIs are derived from direct measurements. The relationship between KPIs do not just exists from level to level but also between KPIs in same level. (Kang et al. 2016)

The KPIs belonging to production group can address single machines or work units, or even the whole production line. Examples of production KPIs are availability and throughput rate. (Kang et al. 2016) The KPIs which belong to quality group are addressing quality or quality performance. For example, scrap ratio is defined as quality KPI and quality buy rate, which is described as the overall percentage of good quality objects after reworks, is an example of quality performance KPI. Moreover, the third group of maintenance KPIs are giving information for setup and maintenance times. Examples of maintenance KPIs are mean time to failure and mean setup time. (Kang et al. 2016)

All the basic KPIs are contributing comprehensive KPIs. The comprehensive KPIs are complex and need more data than basic KPIs. The comprehensive KPIs are presenting the overall performance of the production, and these KPIs may be supported by multiple basic KPIs. OEE is good example of comprehensive KPI because it is giving lots of information about production efficiency, production loss and time usage of single device or group of devices. OEE can use different basic KPIs as a root for the equations. (Kang et al. 2016)

The KPIs cannot be independent because the same raw data and measurements can be used for multiple different KPIs. Various KPIs in distinct categories can have multiple relationships. The Kang et al. (2016) splits the relationships to two different kinds. First, the identity relation of KPIs which is based on definition of KPI. The second type is relevance relation which means that KPIs shares supporting elements.

2.1.5 Utilization Importance in Industry

Fraser (2006) conducted a survey of 135 manufacturers with Industry Directions and MESA International. Manufacturers represent multiple type of industries and manufacturing processes. The survey sorted out how manufacturers use metrics and software systems to improve processes and to support control. The survey was implemented as online survey and was built on multiple questions which were answered by manufacturers. The survey was focusing on 11 preselected KPIs. Research reminds that use of only few KPIs in the survey is quite simplified but should still provide valuable information about how manufacturers use KPIs to improve production.

Based on the responses, Fraser (2006) divides companies to two groups – Business Movers and Others. Business Movers are defined as companies whose have had significant improvements in performance annual over last three years. These improvements can either be dramatic or broad. A dramatic improvement means that a company has improved at least one of the 11 KPIs over 10 %. A broad improvement means that a company managed to improve at least six of the 11 KPIs over 1 %. Common for all the Business Movers is that they represent best practices measuring the performance of the manufacturing. Usually, they have also achieved good operational results which causes improvements in financial performance also.

Other common characteristics of the Business Movers are also identified. For example, the Business Movers are about 50 % more likely to use ADC when gathering data from production than group of Others. KPIs are also displayed to operators much faster. Using of Manufacturing Execution Systems (MES) or dashboards is more common among the Business Movers than among the Others. The Business Movers are more likely to have improvements in quality, customer service, throughput, flexibility, compliance, asset utilization and inventory than the group of Others. Interestingly, the Business Movers answers 6 times more unlikely to question about which KPIs are in use. (Fraser 2006) About this can be deduced that the Business Movers are more aware of production state and status. Fraser (2006) arguments, based on the survey, that companies having effective metrics system are more likely to improve processes and finally gain more market share.

With the help of the survey, Fraser (2006) is also able to detect the most important and widely used KPIs of U.S. manufacturing companies. Because the Occupational Safety and Health Administration (OSHA) is requiring accidental reports from companies, most widely used KPIs are related to safety issues. On-time delivery KPIs, like on-time delivery to customer request and on-time delivery to commit, are the second most used. The next most widely used KPI is manufacturing cycle time. After these come KPIs like overtime, inventory, capacity utilization and OEE.

2.1.6 Quality and Possible Risks

To avoid faulty functions based on poor quality of KPIs, it is important to design KPIs carefully. The blind trust to KPIs is a risk and therefore understanding the whole production process is very important. Experience from manufacturing industry teaches that operators are also manipulating KPIs to give better impression to the supervisors, which makes blind trust to KPIs dangerous. One large company uses KPI that measures the machine pallet loadings during the week. The KPI value is then shown in the dashboard that is located at the top of the workplace. Each week on every Monday, the supervisors of the company take a walk around the factory floor and checks the dashboard values. Because the operators want that the KPI value looks good every Monday, they change the time settings of the dashboard to point to the week when there was lot of loadings done. This causes querying of the measurements to point to the wrong days. The KPI value then looks excellent but it is pointing to the wrong week.

Quality of KPIs is depending on multiple variables. Data gathering speed is often important for KPI to be effective. If data recording takes too much time, the effectivity of KPI can decrease. This can happen for example if data is gathered manually. The latency between measuring the KPI and displaying it to operators and supervisors is affecting the usability of the KPI. On the other hand, only actions taken towards improving the production or processes, based on the measured KPI value, are making the KPI necessary. The KPI is useless if no one is using it because the KPI is not improving the production itself. (Fraser 2006) If the KPI is analyzed too rarely, also the actions are usually done too late. This may cause production lost and ineffectiveness of manufacturing. Linking the manufacturing operations to business KPIs fast enough is also important (Fraser 2006).

Sometimes KPI value can lead to misunderstanding of the behavior of production system, at least if the single machine on the production line is under review. That makes it very important to understand and realize the big picture of the production before making judgments or decisions based on KPI measurements. Mulrane (2016) gives an example of the production line with several machines. If KPI is suggesting that a machine is lacking performance, decision-maker should eye also the machines in front of and after the measured machine. The measured machine could be in the starved state if it is waiting for a machine in front of it. If a machine after the measured machine is faulted, the measured machine might be blocked because it cannot feed the products forward. This means that KPIs cannot be blindly trusted, but it takes a lot of knowledge about the system to really use the KPIs right.

Fitz-Gibbon (1990) proposes that wrongly chosen indicator can even be damaging and gives example where a bad KPI which causes danger. If workers know that some operation is monitored closely, they might get careless with the dangerous event. Choosing the KPI so that the dangerous event is not affecting the value, gives the wrong message to the workers. Noticing that, Fitz-Gibbon (1990) gives one more variable which is affecting the design of KPIs. KPI designer should take account of what kind of messages is KPI given to the people if they know that KPI is collected and how will the people react to it. It is said that you get what you measure. This can be thought as target shifting where manufacturing starts aiming for the better measurement values and therefore the value of KPI is improving. That is one reason why selecting right KPIs is critical for business and why the KPIs are also part of the strategy. For example, if KPIs are measuring scrap amount, it is natural that employers start aiming for smaller waste.

2.1.7 ISO 22400

ISO 22400 is standard for Key Performance Indicators for Manufacturing Operations Management (KPIs for MOM). It will provide the overview of the concepts of KPIs, introduces terminology and describes the methods for KPIs and KPI exchange (ISO 22400:2:2014). The standard will consist four parts, but now there are only two parts published (ISO 22400:2:2014; Johnsson & Kirsch 2014). The parts are listed below.

- Part1: Overview, concepts and terminology
- Part2: Definitions and Descriptions
- Part3: Exchange and use
- Part4: Relationships and dependencies

The title for the standard is Automation systems and integration - Key performance indicators for manufacturing operations management (ISO 22400:2:2014). ISO 22400 defines factory managers, who are responsible of production performance, engineers, who are dealing with process planning of products, manufacturing system designers, software suppliers developing KPIs, and system, device and equipment suppliers as an audience of KPIs. (ISO 22400:2:2014)

KPI is defined by ISO 22400:1 by giving context and content for it. Content is defined as element which is quantifiable and has a unit of measure. The content also includes the formula for the KPI value. The context is confirmable list of conditions that must be met for KPI. ISA 95 and standard IEC 62264 Enterprise-Control System integration are defining the MOM as a set of activities within level 3 of a manufacturing facility. The facility consists the personnel, equipment and material. (Johnsson & Kirsch 2014)

Sometimes the MOM is referred to be MES (ISO 22400:2:2014). The functional hierarchy of manufacturing facility presented in IEC 62264-3 is presented in figure 6.

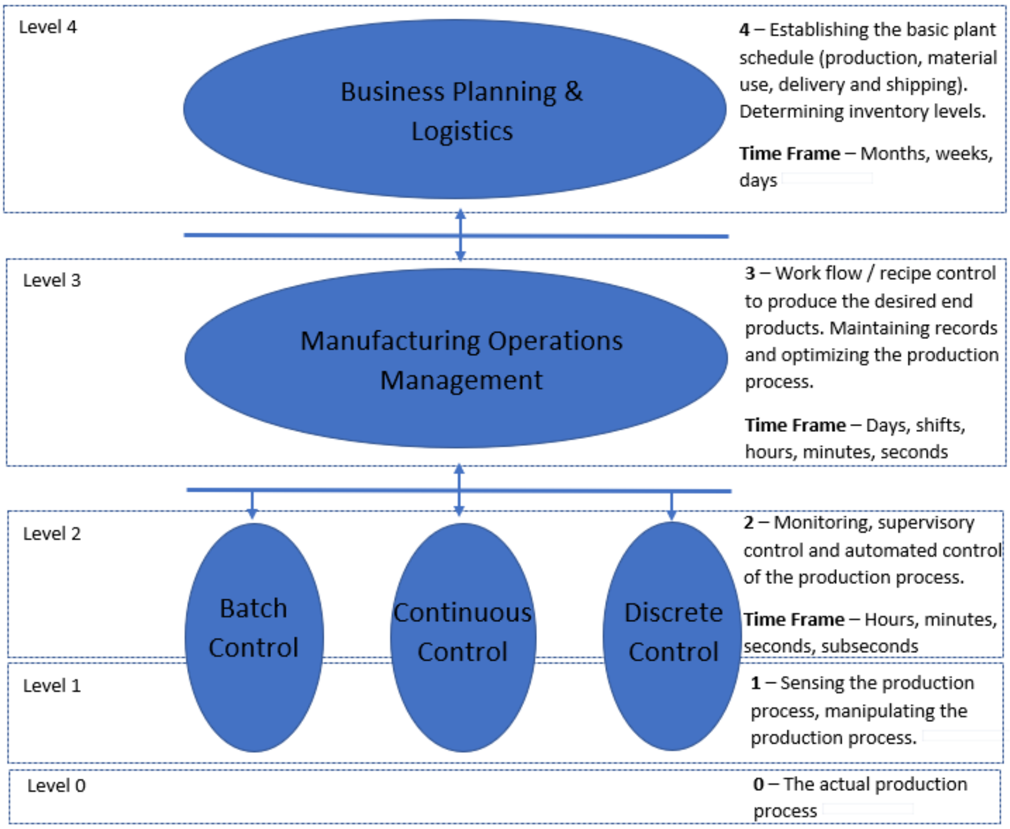


Figure 6. *The functional hierarchy of manufacturing facility based on IEC 62264-3 (ISO 22400:2:2014, adapted from IEC-62264-3)*

The figure demonstrates the 5 levels the functional hierarchy model. The time frame for every level is different and each of them provides distinct functions. The ISO 22400 defines KPIs as a 'residents' of level 3 of the model. The information and measuring values from level 1 and level 2 might be needed to calculate the KPIs and sometimes the KPIs are forwarded to level 4. But most importantly, the KPIs are generated at the level 3 on ISO 22400. There is also multiple type of KPIs at the level 4 and they are related to logistics and business planning, but KPIs at the level 4 are not part of the ISO 22400 which is focusing on manufacturing operations. (ISO 22400:2:2014)

ISO 22400:2 defines 34 KPIs which are designed to be good examples of widely used indicators in manufacturing operations level of industry nowadays. These KPIs can be thought as a palette from where companies can select KPIs which are best fitting and reflecting their purpose. Some of the KPIs are fitting better at discrete manufacturing while other are fitting better for continuous production. These 34 KPIs are presented in table 4.

Table 4. *34 KPIs defined in ISO 22400:2*

Worker Efficiency	Production process ratio	Finished goods ratio
Allocation Ratio	Actual to planned scrap ratio	Integrated goods ratio
Throughput rate	First pass yield	Production loss ratio
Allocation efficiency	Scrap ratio	Storage and transportation loss ratio
Utilization efficiency	Rework ratio	Other loss ratio
Overall equipment effectiveness index	Fall off ratio	Equipment load ratio
Net equipment effectiveness index	Machine capability index	Mean operating time between failures
Availability	Critical machine capability index	Mean time to failure
Effectiveness	Process capability index	Mean time to restoration
Quality Ratio	Critical process capability index	Corrective maintenance ratio
Setup Rate	Comprehensive energy consumption	
Technical efficiency	Inventory turns	

ISO 22400 defines these KPIs with a formula, a time model and an effect model. The formula defines how the numerical value of KPI is derived using measurement data or other output data. Information about physical measurement used in functions, that forms KPIs, is visualized using the time model. Finally, every KPI has own effective model which is like a root-cause diagram. Relationships between KPI and its parameters are emphasized with the picture-like effective model. (Johnsson & Kirsch 2014)

One of the main objectives of ISO 22400 is to define the KPI exchange between MOM applications or between MOM application and another application in business domain.

The KPI exchange and presenting formal UML based KPI template will be discussed in ISO 22400:3. The formal template for KPI is critical for collaboration between different applications. The KPI exchange between application can happen in multiple different ways including event-driven, periodical and demand-based exchange. (Johnsson & Kirsch 2014)

ISO 22400:4 standard will discuss about relationship between KPIs. The relationship can exist when KPIs share the elements used in the formulas which are deriving the KPIs. (Johnsson & Kirsch 2014) Every KPI has own formula and every KPI is calculated differently but some measurement data or other elements can be used to form multiple different KPIs.

Like Fraser (2006) identified in the research, also Johnsson and Kirch (2014) points out that the companies using KPIs, measuring and reporting results and having well informed employers are more likely to improve their financial performance than companies which are not focusing on measurements. Johnsson and Kirch (2014) states that therefore ISO 22400 gives value to the industry. The standard defines widely used KPIs and provides definition for them. This allows adapting the most effective ones for the company production and for the MOMs or MESs.

2.1.7.1 Time Models for Work Units

ISO 22400 defines 4 different type of time models. The time models show difference between planned and actual times. They also demonstrate the different time concepts used in manufacturing environment. First type is designed for work units like devices or machines. The time line model for this is presented in figure 7.

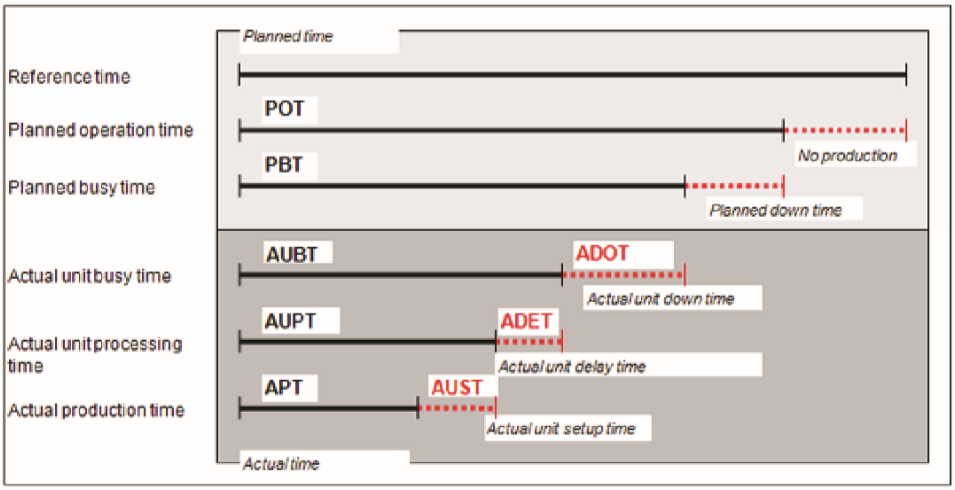


Figure 7. *Time line model for work units, adapted from ISO 22400:2 (2014)*

The time line model visualizes different time concepts and helps to understand why planned time is usually much longer than actual production time. The critical point is in the center of the figure 7, between planned busy time and actual unit busy time. After that point, the time losses are not planned anymore which means unexcepted delays in production schedule. The same kind of model is done for production orders and it is valid when production orders are executed (ISO 22400:2:2014). The time line model for it is presented in figure 8.

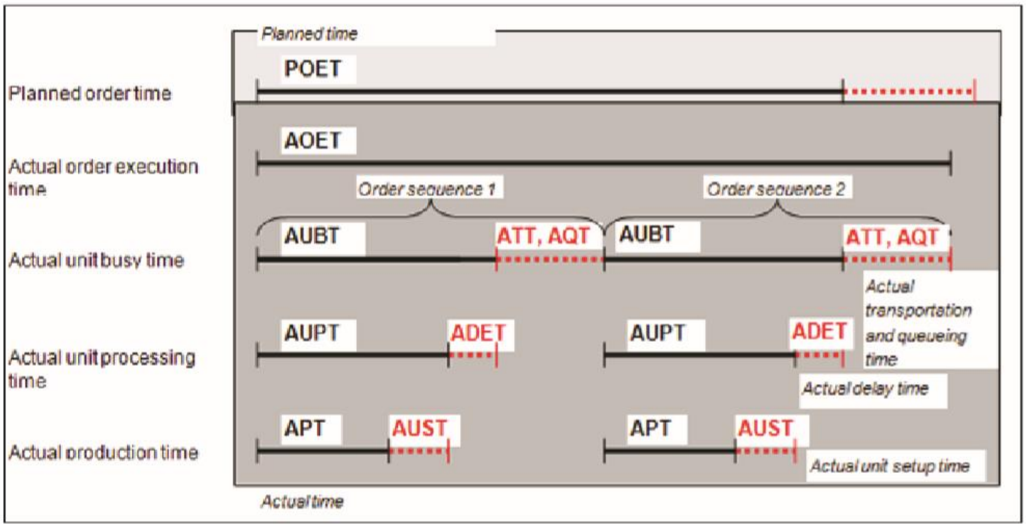


Figure 8. *Time line model for production orders, adapted from ISO 22400:2 (2014)*

The time line model for production orders have multiple occurrences of operations equipment time lines. Multiple separate work units can product the different operations of production order which means that there are multiple work unit time lines in a single production order time line. (ISO 22400:2-2014)

Third of the time models is for personnel. For personnel, the model is simpler than for work units and for production orders, because it has lines for only two different time concepts. These are actual personnel attendance time and actual personnel work time. Coffee pauses, additional breaks or anything outside the actual work time are the differences between the time concepts. The model for personnel is shown in figure 9.

Figure 9. *Time line model for personnel, adapted from ISO 22400:2 (2014)*

ISO 22400:2 also gives an alternative time model for work units for presenting OEE. The model differs from the one presented in figure 7, but the idea is the same. The used time concept elements are the difference between the basic work unit time line model and OEE time line model (ISO-22400:2-2014). The OEE based model for work units is presented in the figure 10.

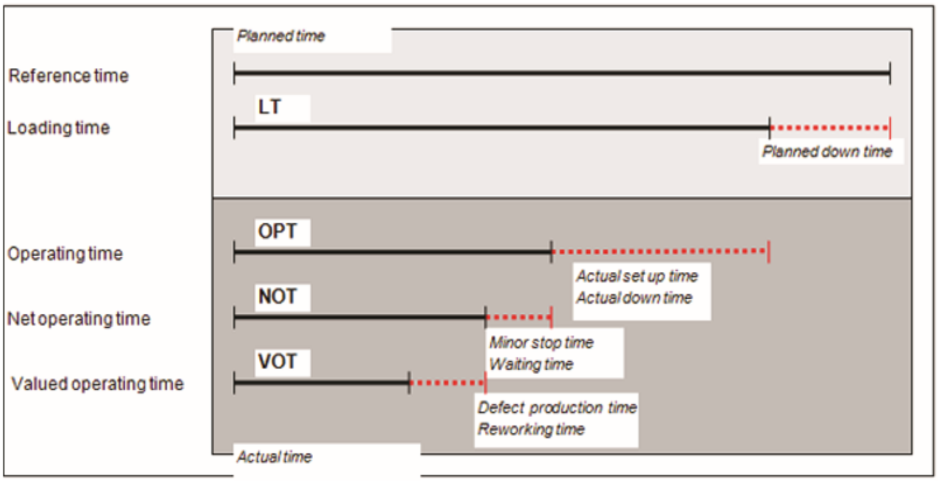


Figure 10. *Time line model for OEE calculation of work units, adapted from ISO 22400:2 (2014)*

The time concept elements in the OEE version of work unit time line model have better support for OEE calculations. Of course, the time line models are not strictly defined, and new time line models can be formed for other KPIs as well.

2.1.7.2 Structure of KPI Description

ISO 22400:2 defines structure for KPI description. The description is presented at the table model where the descriptive elements are on the left side and elements on the right side. The model is presented in the table 5.

Table 5. *Structure of the KPI description (ISO 22400:2-2014)*

KPI Description	
Content	
Name	Name of the KPI
ID	User defined unique identification of the KPI in the user environment
Description	A brief description of the KPI
Scope	Identification of the element that the KPI is relevant form which can be a work unit, work center or production order, product or personnel
Formula	The mathematical formula of the KPI specified in terms of elements
Unit of measure	The basic unit of dimension in which the KPI is expressed
Range	Specifies the upper and lower logical limits of the KPI
Trend	Is the information about the improvement direction, higher is better or lower is better
Context	
Timing	A KPI can be calculated either in - real-time – after each new data acquisition event - on demand – after a specific data selection request - periodically – done at a certain interval, e.g. once per day
Audience	Audience is the user group typically using this KPI. The user groups used in this part of ISO 22400 are - Operators – personnel responsible for the direct operation of the equipment - Supervisors – personnel responsible for directing the activities of the operators - Management – personnel responsible for the overall execution of production
Production methodology	Specifies the production methodology that the KPI is generally applicable for - Discrete - Batch - Continuous
Effect model diagram	The effect model diagram is a graphical representation of the dependencies of the KPI elements that can be used to drill down and understand the source of the element values. NOTE This is a quick analysis which supports rapid efficiency improvement by corrective actions, and thus reduces errors
Notes	Can contain additional information related to the KPI. Typical examples are - Constraints - Usage - Other information

Like table 5 presents, the KPI description gives an overview of the KPI. After reading the description, the understanding about the described KPI is quite comprehensive. For example, KPI availability is presented in KPI description form at the table 6.

Table 6. *KPI description for KPI availability (ISO 22400:2-2014)*

KPI Description	
Content	
Name	Availability
ID	
Description	Availability is the ratio that shows the relation between the actual production time (APT) and the planned busy time (PBT) for a work unit.
Scope	Work unit, product, time period
Formula	Availability = APT/PBT
Unit of measure	%
Range	Min: 0% Max: 100%
Trend	The higher the better
Context	
Timing	On demand, periodically
Audience	Supervisor, management
Production methodology	Discrete, batch, continuous
Effect model diagram	See the figure 11
Notes	Availability indicates how strongly the capacity of a work unit for the production is used in relation to the available capacity. The term availability is also called degree of utilization or capacity factor.

Like can be seen from the table 6, the KPI description gives very good overview for availability KPI. The effect model diagram in the figure 11 describes the structure of the availability KPI.

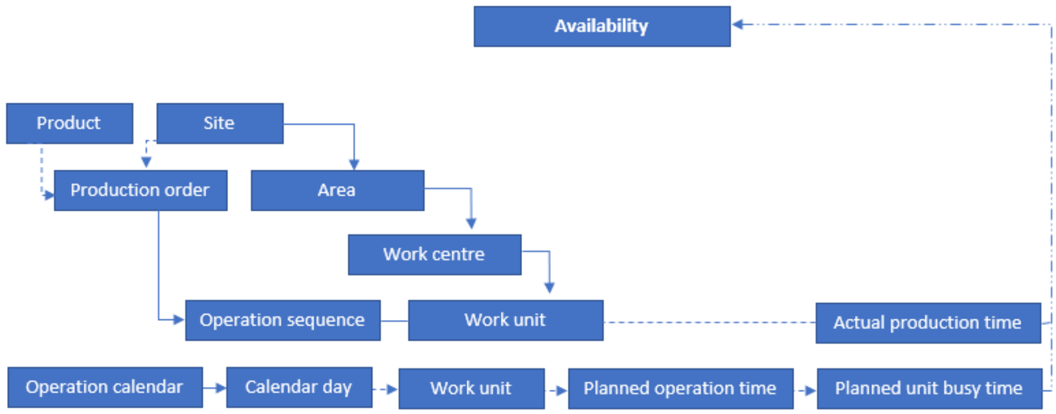


Figure 11. *Effect model diagram of availability, adapter from ISO 22400:2 (2014)*

Like presented in table 6, the formula used for availability is actual production ratio to planned unit busy time. These two components are also part of the effect model diagram. The diagram also shows the component of the actual production time and planned unit busy time. The meaning of the different connectors used in the effect model diagram are presented in the table 7.

Table 7. *Connectors of effect model diagram (ISO 22400:2-2014)*

Line/Arrow	Definition
	Result, through use of a formula, in a KPI
	Includes (1:1 relationship)
	has (i.e. is booked or en-tered)
	consist of (1:n relationship)

2.1.7.3 Criticism

The ISO 22400 tries to be industry neutral, but some researches criticizes the defined KPIs not to be efficient for every industry type. Zhu et al. (2018) made survey for process industry and analyses the ISO 22400 based on survey result and find some gaps between process industry needs and the standard. They say that some of the defined KPIs do not provide useful information for process automation and continuous processes. Couple of KPIs could not even be calculated for continuous process, which make it seems that ISO 22400 is designed only for discrete manufacturing.

Zhu et al. (2018) claims that ISO 22400 is too general in order to implement for all the production types, the KPIs are defined with too clear unit boundaries which makes the standard more applicable to discrete industry. Therefore, Zhu et al. suggest taking process manufacturing features into account when designing KPIs.

2.1.8 Key Performance Indicator Markup Language

MESA International has also developed and defined the standard for key performance indicators. The standard is called Key Performance Indicator Markup Language (KPI-ML) and it is based on XML. KPI-ML standard exploits data models and attributes which are defined in the standards ISO 22400 and ANSI/ISA 95. (MESA 2015)

The KPI-ML defines schemas which makes standard representation for KPI definitions possible. The schemas also specify the KPI instances which are used for specific equipment, operations or devices. Also values of specific KPI instances can be specified with the schemas. The schemas are based on standards ISO 22400:1 and ANSI/ISA-95.00.05-2006. ISO 22400:1 standard is presented in capture 2.3. The name of the ANSI/ISA-95.00.05-2006 is Enterprise-Control System Integration Part 5: Business to Manufacturing Transactions. (MESA 2015)

2.1.8.1 Overview of XML Schema and XML Schema Models

The wanted and allowed elements and attributes of XML documents are defined with XML schemas. The schemas can also restrict contents of the elements and values of the attributes. (MESA 2015; W3Schools 2018) Understanding the importance of the schemas is important because nowadays there are hundreds of XML formats which are standardized and many of them are defined by the XML schemas. The XML schemas can be thought as more powerful alternatives for Document Type Definitions (DTDs). Because XML schemas are based on XML language, schemas can be extended and reused in other schemas. That also allows referencing multiple schemas in one XML document. The XML schemas provide safety for data communication because both sender and receiver know which kind of data should be included in the XML document. (W3Schools 2018) This allows rejecting the messages which are coming in wrong format before even handling them more specially.

There is couple of different widely used models to create XML schemas which includes models like Russian Doll, Salami Slice and Venetian Blind (Cagle & Duckett 2004). KPI-ML schemas follow the model of Venetian Blind (MESA 2015), which has big advantages for reusability. Therefore, Venetian Blind require broad knowledge about which components are globally declared to qualify these components in instance documents (Cagle & Duckett 2004). Immediate child of schema-element is called a global element. The main difference between global and local element is that global elements can be reused in other schemas because they relate to target namespace of the schema. That also means that elements defined in the global namespace can be used as a root for XML instance documents. The Venetian Blind defines only one element, called the root, in global namespace and the local elements use types which are also defined in global namespace. (MedBiquitous 2004)

There are two types of elements which are called simple and complex. Usually types are defined within global namespace and they are used in local elements. In good design principles, using of simple types should be done whenever possible. The simple types can be both restricted and extended. In turn, the complex types can be extended but should not be restricted. (MedBiquitous 2004) For example, the figure 12 presents simple way of using types and elements in Venetian Blind model.

```
<?xml version="1.0"?>
      <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace"https://www.schemaexample.com"
      xmlns="https://www.schemaexample.com">
      <xsd:schema>
         <xsd:simpleType name="IdType">
            <xsd:restriction base="xsd:string">
               <xsd:minLength value="5"/>
            </xsd:restriction>
         </xsd:simpleType>

         <xsd:simpleType name="NameType">
            <xsd:restriction base="xsd:string">
               <xsd:minLength value="3"/>
            </xsd:restriction>
         </xsd:simpleType>

         <xsd:simpleType name="DescriptionType">
            <xsd:restriction base="xsd:string"/>
         </xsd:simpleType>

         <xsd:simpleType name="ValueType">
            <xsd:restriction base="xsd:integer"/>
         </xsd:simpleType>

         <xsd:complexType name="SimpleIndicatorType">
            <xsd:sequence>
               <xsd:element name="Id" type="IdType"/>
               <xsd:element name="Name" type="NameType"/>
               <xsd:element name="Description" type="DescriptionType"/>
               <xsd:element name="Value" type="ValueType"/>
            </xsd:sequence>
         </xsd:complexType>

         <xsd:element name="SimpleIndicator" type="SimpleIndicatorType"/>
      </xsd:schema>
```

Figure 12. *Simplified example of using simple and complex types in Venetian Blind XML-model. XML schema is incomplete and would not work in real environment.*

In the example, XML-element SimpleIndicator, which is type of complex type SimpleIndicatorType, is defined. Venetian Blind hides some complexity of namespace definitions by creating type definitions. In this example, there are 4 simple types and a complex type defined. Simple types IdType and NameType are defined with restrictions. The IdType is type of string and length of it must be at least 5 characters. The NameType is type of string and length of it must be at least 3 characters. Simple types DescriptionType and Valuetype are defined without restrictions. The DescriptionType is type of string and the ValueType is type of integer. The complex type SimpleIndicatorType defines 4 elements which are Id of type IdType, Name of type NameType, Description of type Description-Type and Value of type ValueType. Example on how to form XML document using the schema defined in the figure 12 is given in the figure 13.

```
<?xml version="1.0"?>
<SimpleIndicator xmlns="https://www.schemaexample.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.schemaexample.com/xml/SimpleIndica-
tor.xsd">
      <Id>42452</Id>
      <Name>OEE</Name>
      <Description>Overall  Equipment  Effectiveness  is  percentage  of
      time when production is productive</Description>
      <Value>58</Value>
</SimpleIndicator>
```

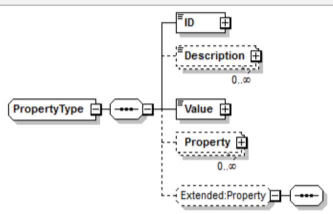
Figure 13. *Example of XML document using schema defined in figure 12*

When using the XML schemas, creating the formal XML documents is straightforward and validating the created XML document is much easier. In the figure 13, the XML document for indicator named OEE is created using the XML schema SimpleIndicator.xsd defined in figure 12. The XML document fulfills all the restricting defined in the schema. The length of Id-element is 5 characters and the length of Name-element is 3 characters. All the needed elements are also defined. The same kind of XML document could also be created for some other indicator, like availability, using the same schema.

2.1.8.2 Key Performance Indicator Object Model

KPI-ML defines the Key Performance Indicator Object Model which defines elements of KPIs. The elements are divided to 2 separate groups which are basic elements and secondary elements. The basic elements are KPI Definition, KPI Instance and KPI Value. Mesa (2015) has also developed schema diagrams for the basic elements, and for some secondary elements like TimeRangeType and PropertyType. The schema diagrams are visualized presentation for schemas and helps to understand the schema content and structure. The secondary elements are presented in table 8. (MESA 2015) The secondary elements are used to construct and define the basic elements.

Table 8. Secondary elements of KPI-ML, adapter from Mesa (2015)

Secondary element	Definition
AudienceType	The excepted audience of KPI
DescriptionType	The description of the element
ID	The identification of element
Name	Human readable identification of the element
Production-Methodol-ogyType	Identifies the measurement type of the element. Can be standard type or special type. Standard types are continuous, batch, discrete and other.
PropertyType	
RangeType	The range for the KPI instance. Normally mathematical limit like 0% -100%.
ResourceRefer-enceType	The reference to a resource. Standard resource types are listed in ISA 95.02.
ScopeType	Identifies the resource element for which the KPI is relevant for. Examples are work unit, product and personnel.
TrendType	Defines how values are read. For example, higher is better or lower is better.
TimeRangeType	Defines the time range of the elements. For example start of end time, start and end time or set of time intervals.
TimingType	Defines how often KPI is calculated. Standard types are real-time, periodically or on-demand and other.
IdentifierType	Id of other KPI definition or KPI instance which is used for calculating the current KPI.

The KPI Definition is defining indicator and contains general information for KPI including, for example, name, range, trend and formula. The definition is demonstrated in ISO 22400:2. The schema diagram of the KPI Definition element is presented in the figure 14.

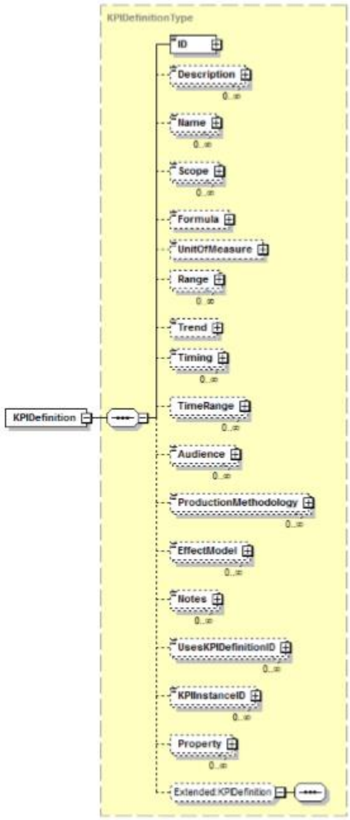
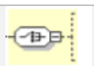
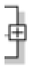


Figure 14. *KPI Definition schema diagram, adapter from MESA (2015)*

As can be seen, the KPI definition is constructed using secondary elements defined in ISO 22400:2. The KPI element KPIDefinition is built with fixed order of elements. Some of the secondary elements can be used only once in the definition, but some of them may occur multiple times. Some of them are optional and therefore they do not need to exist in the definition. The conventions used in the schema diagram are demonstrated in table 9.

Table 9. Schema diagram convention (based on Mesa 2015)

Convention	Definition
	Name of the element or element type
	Fixed order of elements
	One instance only
	One to many instances
	Zero or one instance
	Zero to many instances
	Selection of alternatives
	Contained elements or attributes
	No contained elements or attributes

Based on the KPI definition schema diagram and conventions for it, can be seen that KPI definition have to have one instance of ID but all the other secondary elements are optional. Mesa (2015) provides schema diagrams also for KPI Instances and for KPI Value. These are presented in the figures 15 and 16.

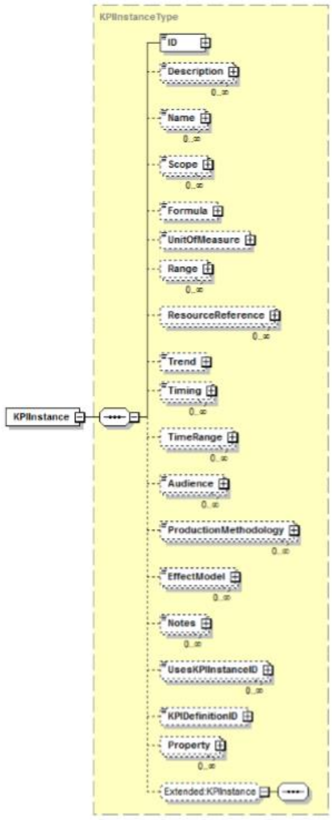


Figure 15. *KPI Instance schema diagram, adapted from Mesa (2015)*

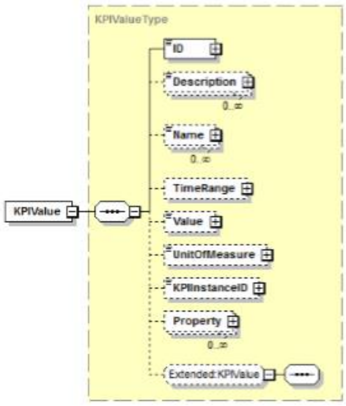


Figure 16. *KPI Value schema diagram, adapted from Mesa (2015)*

The KPI instance is a specimen for KPI definition. The instance is specified for specific resource like site, work place or machine. The definition is like base for all the same KPI instances. This means that one KPI definition can be linked to multiple instances. The KPI value is defining specific value for the KPI instance. The value has secondary elements like name, time range and unit of measure. Again, one KPI instance may be connected to multiple KPI values. Every three of these basic elements can also have extra properties which helps defining specific models. For example, extra properties for KPI value can be formula and values used for the it (MESA 2015). The basic elements and their relationships in the object model are presented in figure 17.

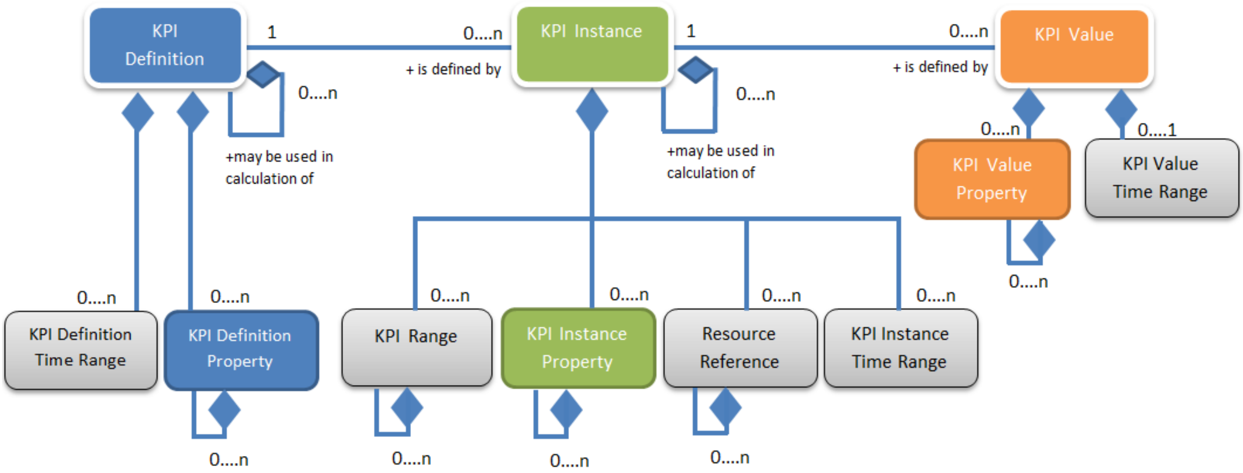


Figure 17. *KPI Object Model used in KPI-ML, adapter from Mesa (2015)*

The KPI object model in KPI-ML is derived from the object model of ISO 22400:1 with multiple improvements and changes. One of the main differences to the ISO 22400:1 object model is that KPI Values may have properties which are optional and are mainly used for information exchange. Secondly, properties of KPI definitions, instances and values can be recursive. Property definitions in the ISA 95.02, IEC 62264:2 and MESA B2MML are consistent with the properties in KPI-ML. ISO 8601 formatted time ranges is also added to KPI-ML to specify time ranges and intervals inside ranges. One of the major modifications is that hierarchy used in ISO 22400:1 and ISO 22400:2, where lower level KPIs are used to calculate higher level KPIs, is presented in KPI-ML with elements called UsesKPIDefinitionID and UsesKPIInstanceID. This hierarchy is used as “may be used in calculation” association in ISO 22400.

The example of KPI-ML is provided in figure 18, where the KPI definition for Availability is formed. The description based on ISO 22400 for Availability is provided in table 6.

```
<KPIDefinition    xsi:schemaLocation=http://www.mesa.org/xml/KPI-ML-V01
KPI-ML-V01.xsd              xmlns="http://www.mesa.org/xml/KPI-ML-V01"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <ID>A100</ID>
    <Description>
      Availability is the ratio that shows the relation between the   ac-
      tual production time (APT) and the planned busy time (PBT) for a
      work unit.
    </Description>
    <Name>Availability</Name>
    <Scope>Work unit</Scope>
    <Scope>Product</Scope>
    <Scope>Time period</Scope>
    <Formula>Availability = APT/PBT</Formula>
    <UnitOfMeasure>%</UnitOfMeasure>
    <Range>
        <ID>Natural</ID>
        <Description>Natural Range</Description>
        <LowerLimit>0</LowerLimit>
        <UpperLimit>100</UpperLimit>
    </Range>
    <Trend>Higher-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <ProductionMethodology>Continuous</ProductionMethodology>
    <Notes>
      Availability indicates how strongly the capacity of a work unit
      for the production is used in relation to the available capacity.
      The term availability is also called degree of utilization or ca-
      pacity factor.
    </Notes>
</KPIDefinition>
```

Figure 18. *KPI-ML description for KPI Availability based on ISO 22400*

As can be seen, the XML file contains all the same elements than KPI description of ISO 22400. If some element has multiple values, the element is listed multiple times. This is done with Scope, Timing, Audience and ProductionMethodology -elements. The XML files of KPI instance and KPI value are following the same XML principles.

2.1.8.3 Transaction Definitions

The XML schemas and definitions create base for transactions between applications and this is one of the main areas of KPI-ML which provides the supportive set of elements for transactions. The transactions are defined in the ISA 95 Part 5 Business-to-Manufacturing Transaction standard. OAGiS 9.6 model is followed by KPI-ML for transaction messages which use the OAGiS XML schema structure. Data objects relate to ISO 22400 and are KPI-ML elements. (Mesa 2015)

The transaction messages are built with 2 different kind of objects, so called verbs and nouns. Data objects which are targets of the actions are defined as nouns, while actions and responses to the actions are defined by verbs. The message name is combination of the verb and the noun. For example, the noun “KPIValue” can be queried with the verb “Get” which is forming an element called “GetKPIValue”. (Mesa 2015)

Transaction elements are constructed using 2 different elements which are Application-Area and DataArea. The ApplicationArea elements is similar for all the transaction elements while DataArea elements are unique. The DataArea element is divided to 2 parts. The first part is the action, the verb. The second part defines the exchange element, the noun. figure 19 presents the transaction element as XML-format and figure 20 presents the transaction element in visualized form. (Mesa 2015)

```
<GetKPIValue … releaseID="KPI-ML-V01">
    <ApplicationArea>
        …
    </ApplicationArea>
    <DataArea>
        <Get>
           …
      </Get>
      <KPIValue>
           …
        </KPIValue >
    </DataArea>
</GetKPIValue>
```

Figure 19. *Example of empty transaction element XML adapted from Mesa (2015).*

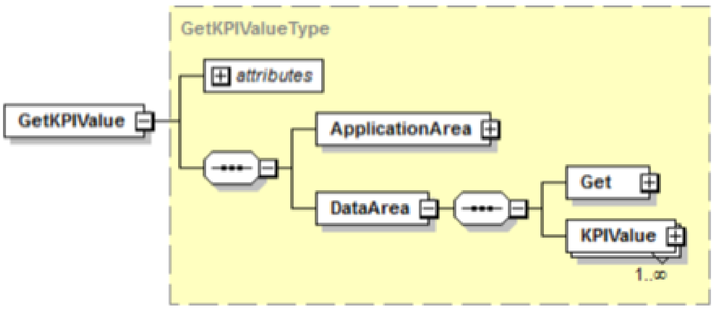


Figure 20. *Transaction element structure visualized, adapted from Mesa (2015)*

Faulty or successful processing of the message can be identified with confirmation message. The confirmation message is always type of ConfirmBOD, where BOD stands for Business Object Document and it is following OAGiS. The message may define in ApplicationArea if the confirmation for the message is needed always, in case of error or never. Mesa (2015) suggest using confirmation messages only for critical messages and CANCEL messages because otherwise the number of messages can be huge. Also, the GET message already have SHOW message as an answer, PROCESS message has on ACKNOWLEDGE message and CHANGE has RESPOND message responding to them. (Mesa 2015)

KPI-ML defines 3 various transaction models which are push model, pull model and publish model. In the pull model, GET verb is used by data user to request the data and SHOW verb is used by data provider to respond the data to the user. The pull transaction model is sometimes called GET/SHOW data exchange. (Mesa 2015) The figure 21 shows example of transaction with pull model.

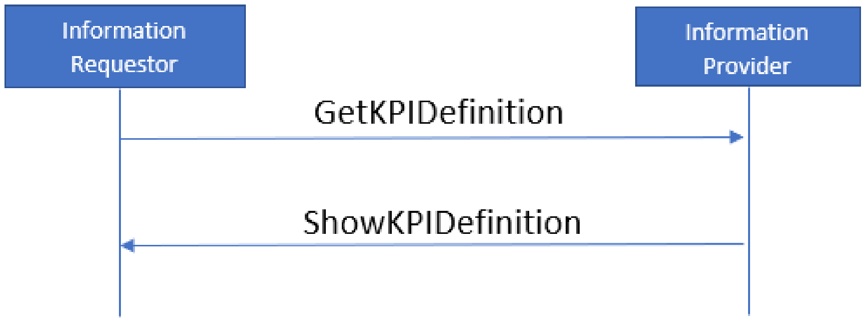


Figure 21. *The transaction using pull mode, so called GET/SHOW data exchange based on Mesa (2015)*

The GET verb is combined with KPI object to request right kind of data. The additional attributes are used for different objects. The identification ID is used for every object, but for example start and end time only for KPI Values. Also, the SHOW responses are different for distinct objects. Every SHOW response has attribute for action code which is either accepted or rejected, but the data is different for different KPI objects. When requesting data for specified KPI Definition using GetKPIDefinition, the ShowKPIDefinition response shows data for all properties and their attributes of the requested KPI. Also, IDs of all the KPI Instances linked to KPI Definition are listed in the response. (Mesa 2015)

The SHOW response for GetKPIInstance includes attributes and properties of the instance, while the SHOW response for GetKPIValue includes values for specific KPI value. The values are following the rules specified in GET. These are for example time range and unit of measure. (Mesa 2015)

PROCESS/ACKNOWLEDGE, CHANGE/RESPOND, and CANCEL messages are used in the push transaction model. In the push transaction model, the other application is processing, changing or cancelling the data of the data owner. The data owner, also called information receiver, can respond with CONFIRM and ACKNOWLEDGE transactions but usually only ACKNOWLEDGE is used. Example of PROCESS/ACKNOWLEDGE transaction is presented in figure 22. (Mesa 2015)

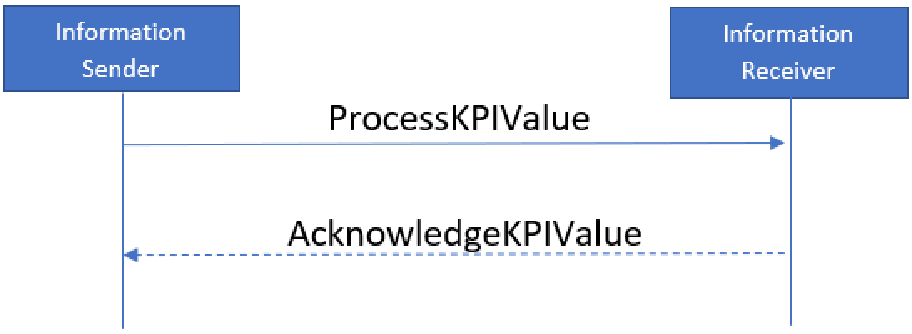


Figure 22. *PROCESS/ACKNOWLEDGE transaction example using ProcessKPIValue message based on Mesa (2015).*

In the PROCESS/ACKNOWLEDGE transaction, the information sender is sending process transaction to the information receiver. The receiver will process the data and acknowledge the results to the information sender. The CHANGE/RESPONSE transaction works similarly to PROCESS/ACKNOWLEDGE transaction. The figure 23 presents this. (Mesa 2015)

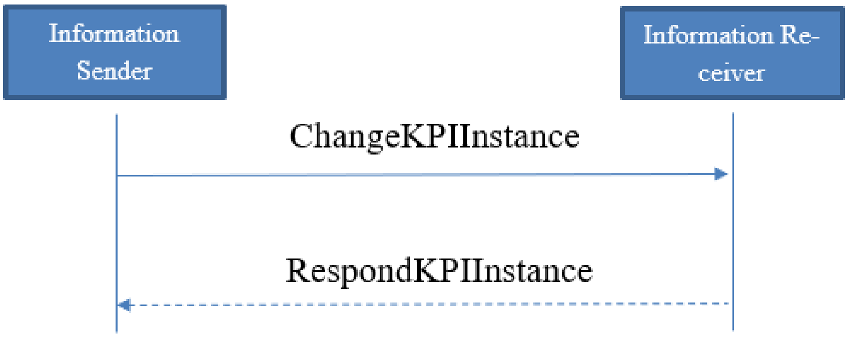


Figure 23. *CHANGE/RESPONSE transaction example using ChangeKPIInstance message based on Mesa (2015).*

With the CANCEL transaction, the information sender can cancel the data of KPI objects. The CANCEL message may not be responded but the information receiver may send confirmation when cancelling function is executed. The example of CANCEL transaction can be seen in figure 24. (Mesa 2015)

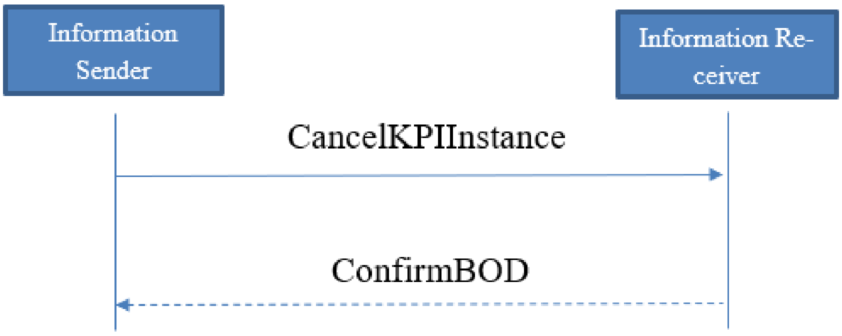


Figure 24. *CANCEL transaction example using CancelKPIInstance message based on Mesa (2015)*

The third transaction model used in KPI-ML is publish transaction model. The verbs ADD, CHANGE and DELETE are used within the publish transaction model. They are determining which kind of functions are taken on the published data. The ADD-verb is normally used to publish if new KPI objects are introduced. There can be multiple subscribers for the published data but in the figure 25 only one is presented. (Mesa 2015)

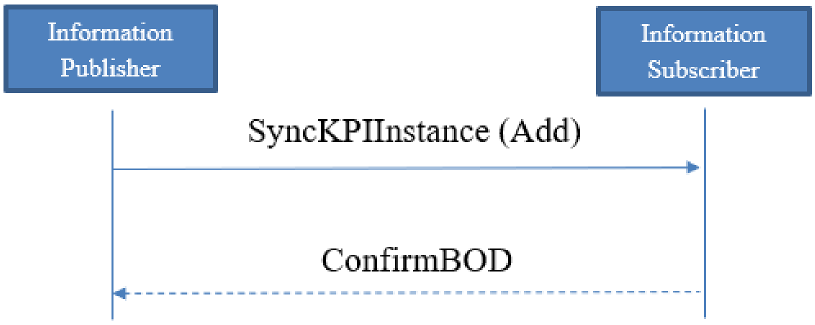


Figure 25. *Publish transaction using ADD-verb based on Mesa (2015)*

In the figure 25, the publish transaction is used to inform about new KPI instance. The confirmation response can be used with publish transaction but normally it is not used. The CHANGE verb is used for publishing information about KPI object change. Like every publish transaction, also this can be sent to multiple subscribers. The figure 26 demonstrates the usage of CHANGE verb with one a subscriber.

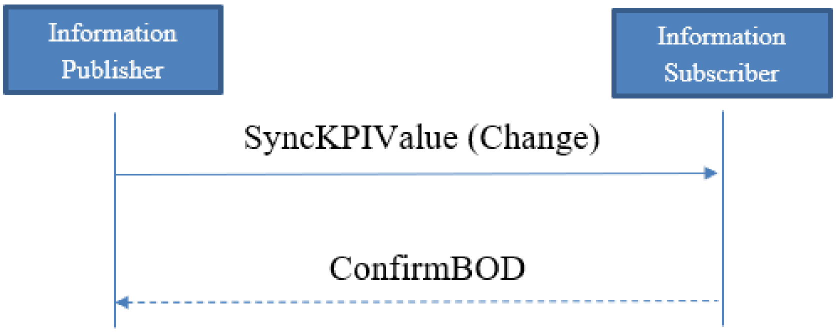


Figure 26. *Publish transaction using CHANGE-verb based on Mesa (2015)*

In the figure 26, the publish transaction used to inform about KPI value change. As usually with publish transactions, the subscriber can send confirmation. The DELETE verb may be used to inform subscribers about KPI object deletion. (Mesa 2015) The example in figure 27 presents the publish transaction which is send only to one subscriber.

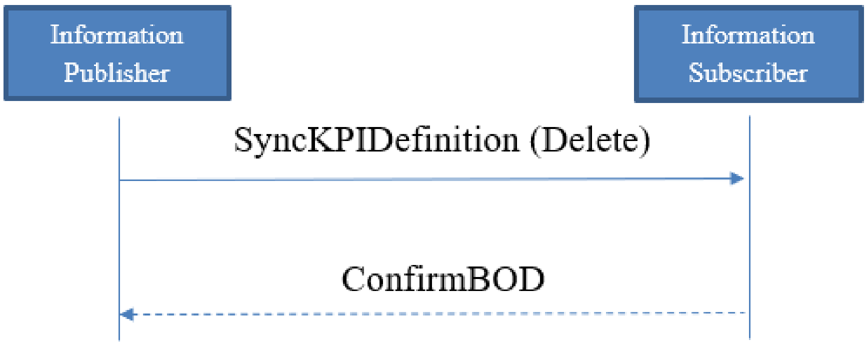


Figure 27. *Publish transaction using DELETE-verb based on Mesa (2015)*

In the example in figure 27, the publish transaction is used to publish information about KPI definition deletion.

For the project of this thesis, the transactions defined in KPI-ML are not so useful because the KPI values are mainly handled within the Inspector. Hopefully in the future, Inspector is integrated with customer other applications, like ERP or MES, and then standardized structure of KPI transaction will be effective and useful. Possible screens and panels for operators could also receive the information about KPIs with KPI transactions.

2.1.9 Selecting and Implementing Right Key Performance Indicators

The manufacturing companies around the world have vast number of different indicators which are reflecting the production efficiency, speed or anything man can even think of measuring. That makes it truly important to select the right KPIs for the manufacturer's business (Fraser 2006; CA 2015). Selected KPIs should cover all important aspects of manufacturing process making the indicator selection balanced (Rakar et al. 2004). This chapter presents couple of different models for KPI development and discuss about KPI selection.

Three-level hierarchical structure for performance indicators (Rakar et al. 2004) defines three different levels sorted by their importance. Each level presents various requirements for production or process, and KPIs should be developed so that requirements of first level can be ensured before starting development of the KPIs for next level. The levels can be seen from figure 28, which is based on figure in Rakar et al. (2004).

Figure 28. *Three-level KPI framework based on Rakar et al. (2004)*

Requirements, and therefore indicators for measuring these requirements, included in the first level belongs to the safety and the environment of the company. In this concept, the environment means how company conformance different regulations and standards. Usually the production cannot be started before the company fulfills all the regulations and safety requirements, so this makes implementing monitors and KPIs for the first level the highest priority. Examples of the environment KPIs are fresh water consumption per product and waste generated per product, and for safety KPIs number of accidents and number of hazardous alarms. (Rakar et al. 2004)

The second level has requirements for production tracking and measuring. KPIs related to quality, efficiency and production plan tracking are developed at this level. The third level consist requirements related to the employees of the company. It may consist for example KPIs for number of sick leaves or number of innovations proposed by employees. Because the third level requirements are often related to the environment and production, KPIs for that level are usually implemented last. (Rakar et al. 2004)

While importance of this three-level structure comes by the ideal situation where company starts implementing indicators from the simplest and the most mandatory ones towards more complex and special ones (Rakar et al. 2004), it is not always the case for companies. Sometimes the companies may be interested only about defining and implementing indicators for production measurements to improve performance and may miss the possibility to use the KPIs also for safety and job satisfaction measures. But should be considered, that some companies might use some performance indicators for the environment or safety without noticing or realizing that KPI is developed.

Measuring the performance is not remaining static which means that also KPIs should be developed and changed. Because KPIs are one of the key factors allowing continues improvement in production, improving and changing KPIs whenever production is changed is very relevant for success (Fraser 2006). Technologies, marketing areas and strategies, and reasons to measure performance varies over time, which causes a need to improve used KPIs (Effendi et al. 2014). The change of the industry environment can happen rapidly so changing the KPIs might need to be done surprisingly soon (Ofori-Boateng 2017). Sometimes companies may use huge amount of resources for searching of perfect set of indicators without realizing that there might not be the perfect set. Therefore, developing the right KPIs should be thought as a dynamic process. When the company has found a working set of indicators, review process for them is also needed ensuring indicators effectiveness also in the future. (Keeble et al. 2003)

2.1.10 8-step Iterative Closed Loop Model and CI Procedure

Rakar et al. (2004) introduces 8-step iterative closed-loop model, which is based on model of Bennett et al. (1999), for developing and deriving KPIs from the production. Way to define, develop, measure and report KPIs with closed-loop model is presented in figure 29, which is constructed based on figure of Rakar et al. (2004). As can be seen, defining KPIs is continues progress, and sticking with the same production targets, and therefore KPIs, is not possible usually. That is why effective methods to define and to develop KPIs continuously are needed for efficient usage of KPIs to improve production and report current performance related to goals and targets.

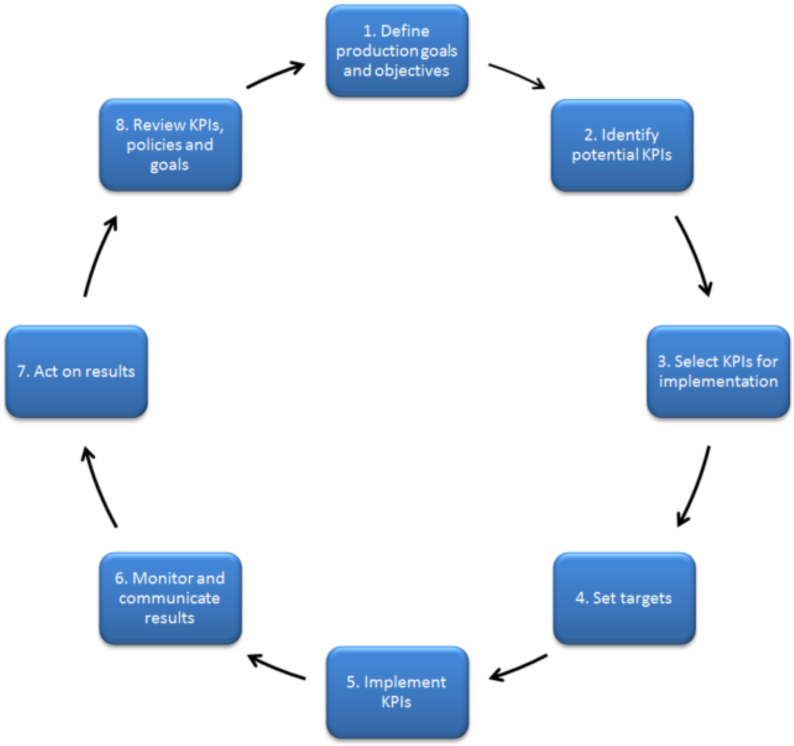


Figure 29. *8-step iterative closed loop model for KPI development based on Rakar et al. (2004)*

In the first step the company needs to define goals and targets for the production which are projecting extensively the mission of the company. The fine defined goals are needed for effective identification of potential key performance indicators, because the KPIs should also reflect the targets of the production. Rakar et al. (2004) suggest that companies should identify as many core indicators as possible in the second step. However, some other literature, for example Fraser (2006), proposes that companies should concentrate for only a few performance indicators. Fraser (2006) also claims, based on phone interviews with different manufacturers, that using just a few frequently reviewed and refined KPIs is effective for the companies. Anyhow, identifying the most important indicators of production should be done at the step 2.

After the second step there should be group of potential KPIs which are then evaluated and reviewed in the third step. To ensure data availability and to motivate employees, employees should be included in the KPI selection step as much as possible. In the third step, also some specific indicators outside the core indictors defined in the step 2 could be selected. (Rakar et al. 2004)

The targets for KPIs are set on the fourth step to ensure management commitment, while the implementation of the KPIs are done at the fifth step. Implementing KPIs can take a lot of time because data gathering, value calculating and result evaluating are time consuming operations. In the next step, the KPIs are monitored and communicated to management, employees and other interested groups like stakeholders or sister companies. Presenting the KPIs also for public parties may help to improve trust and image of the company. (Rakar et al. 2004)

To improve the performance of the production continuously, the acts on results of KPIs are needed. Because this 8-step model focuses on KPI development, the acts are made to improve measurements and calculations. Of course, in the manufacturing environment, also acts on production improvement must be made based on KPI results. The reviews for KPIs, goals and policies are made in the last step of the model. The life time of the KPI ends here before setting new goals and selecting new indicators to support these new targets. (Rakar et al. 2004)

Kang et al. (2016) defines CI procedure for KPI development. Part of it also form a loop like 8-step iterative closed loop model. The CI procedure is concentrating more on KPI development instead of defining KPIs based on production goals. But overall, both models are targeting continuous development of KPIs. The CI procedure can be seen in figure 30.

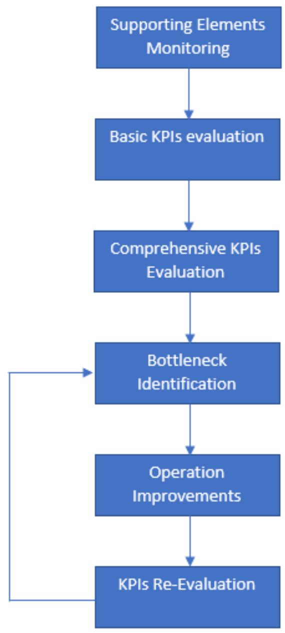


Figure 30. *The CI Procedure for KPI development based on Kang et al. (2016)*

First, the measurement and monitoring are designed to get the raw data. This data is then used for designing basic KPIs which can be used for comprehensive KPIs development. After that starts the KPI results analyze by detecting the bottlenecks of production. When operations are done for improving the manufacturing system and removing bottlenecks, KPIs must be re-evaluated. If there is need for change or modification of KPIs, it can be done. Then the same cycle repeats from the bottleneck identification. The target is to continuously identify and remove the bottlenecks by developing and defining as good

KPIs as possible. The idea is the same than with 8-step iterative closed loop model – to always improve the used KPIs to gain more value for them.

Before implementing a key performance indicator, the structure of the indicator must be defined. Rakar et al. (2004) suggest defining four key properties for each of the indicator. The first property defines the unit of measurement. For example, the unit can be euros, percent or centimeters. The second property indicates if the type of the measurement is absolute or adjusted. The absolute type means that the total amount is measured, for example how much water is used in the factory per year. “How much water is used by a unit per year” is an example of an adjusted type indicator. Period of measurement is the third property defined. For example, the period can be daily, monthly or yearly. The fourth indicator reflects the boundaries of the indicator. The boundary means how far the measurements are going in the environment. That means defining, if the measurements are for the production line or for the whole manufacturing system or perhaps even for the suppliers and distributors.

To summarize and simplify, the good KPI should be clear enough to be used effectively. It should point the operators to the right direction and motivate them to do their best. Moreover, the KPI could point out the targets and goals, not only measure the work. Good KPI should also be comparable to the older values to be able to detect if the performance is getting better. Finally, good KPI should also be reviewed time to time, just to keep it effective also in the changing environment.

2.2 Production Flow Analyse and Monitoring in Discrete Automation Industry

This chapter focuses on production flow monitoring and traceability. Nowadays, traceability of production items is getting more and more important. In case of item breaks or faults, it is important to be able to track the route of the item in production process. It is not easy to achieve traceability in every industry type and therefore, also short preview to process automation is provided. InSolution is not closing doors from any customer and therefore, knowledge about flow monitoring generally is important. Different process layouts are also presented because traceability data is closely linked to layout designing.

2.2.1 Production Flow Monitoring

The manufacturing industry is typically divided to two main types based on the how the ready products are realized. These two types are process and discrete industry and they have different characteristic. Process industry is sometimes called continuous industry because of the manufacturing process. Third industry type, hybrid industry, can also be identified. Briefly, the term hybrid industry can be used when continuous production is controlled with discrete control. The differences between process and discrete industry and the flow monitoring differences are discussed below.

2.2.1.1 Differences between Discrete and Process Industry

In the process industry, the ready product is constructed during continuous or batch production using formulas and recipes (Müller & Oehm 2018). The raw material can be though as ingredient in that formula. The ready product is something which cannot be converted back to the raw material because often the raw materials are liquids or fluids, or the process is mixing multiple materials together (Zhu et al. 2018). Sometimes production is running with batches rather than continuously if same plant is used for different products. Typical examples of process industry are chemical, pharmaceutical and food industry. The operator role in process industry is proactive and includes lot of problem solving (Müller & Oehm 2018).

The process industry plants are usually large and the main control in process industry are usually valves. Other common components are pumps, reactors and heaters. Highly designed control strategies are used to control streams of mass and energy to allow raw materials transformation to ready product using chemical and physical laws. Therefore, process is closed system which is controlled with system specific parameters, for example for temperature and pressure. The process industry is usually highly automated. (Müller & Oehm 2018)

The discrete industry produces ready products with discrete states and transforms raw material into discrete units. Discrete processing can also be divided to two main types, which are continuous and intermittent (Müller & Oehm 2018). The assembly production, where multiple work pieces are combined to get a ready product, is also common in discrete automation. Products are assembled with bill of materials (BOM) rather than formulas. In discrete industry, the materials, work flows and ready products are varying from order to order. Therefore, discrete production is often based on production orders. Sometimes, the automation control system of the process is also functioning based on production orders. In discrete automation the ready material can often be break back to the raw material. The operator work is to ensure quality, solve faults and keep production running (Müller & Oehm 2018).

The scale of the discrete industry plants is usually from small to medium size. Usually the production is combination of almost individually working machines or devices where every machine or machine group works in specific operations (Müller & Oehm 2018). The plant can consist of machines like milling machine, washing machine, CMM, wrapping machine or engraving laser machine. The discrete plant is open system making environmental distractions, like temperature or dust, to affect behaviour of process. The automation level of discrete industry is varying a lot between companies, but for example in US food manufacturing, the automation level is medium to high. (Müller & Oehm 2018) The table 10 presents main differences between process and discrete automation.

Table 10. *Some differences between process and discrete industry based on Müller & Oehm (2018)*

	Process industry	**Discrete industry**
Scale	Large	Small to medium
Equipment	General purpose equipment	Dedicated machines
Distributed processes	Tightly coupled, plant-wide control strategies	Coordinated by the flow of discrete parts
Process parameters	Temperature, flow, pressure, level, weight	Force, temperature, time
System	Closed	Open
Level of automation	High	Medium to high
Typical unit operations	Distillation, crystallization, mechanical separation, chemical reactions	Forming, separating, joining, dispensing, and moving
Time criticality	Not fast but at the right time	High
Production pausing or stopping	Problematic	Possible
Product shape	Undifferentiated mass or fluid	Three-dimensional objects of different materials
Product value	Often high	Often low

Like table 10 and earlier observations points out, the differences between two main industry types are quite huge. But regardless of all the differences between manufacturing types, the main idea with all the manufacturing processes is to convert material into products by adding value to material during different process steps (van Eekelen 2008).

2.2.1.2 Production Flow in Different Production Types

The verb "flow" is defined by van Eekelen (2008) with phrase "describes how materials and information are processed". The differences between process and discrete industry are big, which also means that production flow characteristic is diverse in these. Three different kind of production flows can be identified: job-shop, flow line and continuous flow (van Eekelen 2008).

Like earlier mentioned, in continuous production material flows, literally, through factory with very strict route. On the other hand, in discrete automation, the flow line is discrete manufacturing, but the route of the material is fixed. These kind of flow lines exists typically in industries where the production volume is very high, and the variation of the single product is standardized. Examples of these exists in part of the car industry, in food industry and in mass production industry. Re-entrant flow line is a term for flow line where material needs to pass some parts of the line multiple times. (van Eekelen 2008)

The main interest of this thesis is in job-shops. In the job-shop type of production, the material flows inside the factory without any strict route and the amount of different, very customized products, is large. For example, some industries producing custom furniture can be using job-shop type of production. (van Eekelen 2008) The main differences between flow line and a job-shop production is presented in the figure 31. The material flow direction is called downstream and the opposite direction is called upstream.

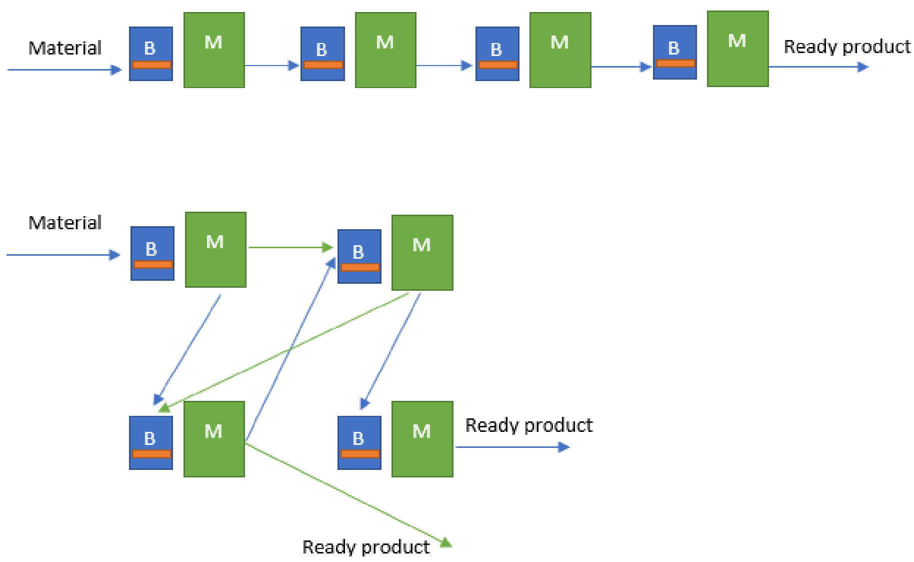


Figure 31. *The upper line is a flow line where material flows from machine to machine following a strict route. The below line presents job-shop production. There blue arrows presents one type of product and green arrows other type of product. B presents buffers and M machines.*

Often manufacturing floor consist multiple workstations which are combination of machines and buffers. The machines process material from the buffer and sends it to the buffer of next workstation. The figure 31 presents buffers with a letter B and machines with a letter M. The machine can be single-lot machine, which is producing single material part at a time, or a batch machine, which is producing multiple parts from the buffer simultaneously and sending multiple parts to the other workstation after the production step is finished. (van Eekelen 2008)

Understandable, monitoring of the production flow varies highly in process and in discrete automation industry. Nevertheless, complexity of monitoring system can become high when both physical product data and flow information is monitored. However, implementing a monitoring system is important because problems with scheduling, control, planning or quality can occur by missing flow information. (Chongwatpol & Sharda 2012) Sometimes traceability in production is though as combination of process information and production flow data. This is achieved with traceability systems which are linking together the product and process data. (Kvarnström 2008)

2.2.2 Traceability as a Production Flow Monitoring

Flow monitoring is often linked with term traceability in literature when discrete production, especially job shop production, is monitored. Traceability can be though as knowledge about everything what happens to product from raw material to ready product through production process, including for example used machines, tools and operators (Nair & Shah 2007). On the other hand, traceability allows tracing production back to the root causes of possible faults or quality issues with the products allowing continuous improvement (Töyrylä 1999; Kvarnström 2008). ISO 9000 (2005) standard presents traceability as ability to track location, history or application for item of interest.

Kvanström (2008) argues that traceability of products is very important part of quality management systems. Also, Töyrylä (1999) mentions that need for higher quality and better logistic management is increasing the need for traceability. Just to mention some, traceability of products can also answer to the need of activities like Risk Management, Logistical Flow, Information Management and Commercial advantage (Töyrylä 1999).

Of course, the traceability is important for most of the industries, but the relevance of the traceability can be identified with five different factors which are presented for example in study of Töyrylä (1999). These factors are, more or less, linked with the production items which, again, links traceability to discrete automation. These factors are item value, item criticality, item life time, system complexity and external environment (Töyrylä 1999). Higher item value is argued to be reason for better traceability for item. The item, which can cause injuries, safety issues or system damages while broken, is thought to be crucial and therefore it should be traced better. The longer life time of the item is causing higher need for traceability due to more items in use. Again, the system complexity increases the traceability needs because the products complexity is thought be higher also.

The external environment, like legal environment or customer values, are also affecting to the need of traceability.

To summarize these five key factors, 2 points for traceability implementation decision can be found. The first point is a cost-benefit analysis which works just like any other investment in the industry. This means, that costs of the traceability implementation are compared with the benefits which are thought to be gained. The second point is direct requirements coming from customers, authorities and laws. (Töyrylä 1999) These points can be used to determine the level of traceability needed.

The interest in traceability in process automation arise within food industry after EU regulations caused by mad-cow disease (Töyrylä 1999). But as earlier mentioned, traceability is mostly linked to discrete production and achieving traceability with continuous processes is challenging. The challenge is caused by missing trackable units, like batches or products, in continuously and unstoppable flowing processes. Also, the resolution of traceability systems in continuous and discrete production differs dramatically. Material mixing, and reflux flows are also disturbing trackability in process automation. (Kvarnström 2008)

Even it is challenging to achieve traceability in process automation, Kvarnström (2008) was able collects different flow monitoring techniques for process automation from literature. These are for example chemical tracer, radioactive tracer, coloring agent and magnetic tracer. The tracers are used by dropping these to continuous flow at regular time intervals. With this method, imaginary or virtual batches can be created to the production flow which allows implementing batch-technique traceability. However, Kvarnström (2008) wants to notify that the benefits for traceability in process automation are not so known.

2.2.2.1 Restricted and Extensive Traceability

Traceability in manufacturing can be divided to the 2 different types which are separated by the usage of tracking data. These are called restricted and extensive traceability (van Dorp 2002). Even though von Dorp (2002) uses the terms mostly for supple chain monitoring, the use can be expanded to the manufacturing plant monitoring, for example for job-shop plants.

The restricted traceability means that products are tracked to gain historic results of recorded identification. This creates visibility to product timeline and possibility for backward and forward tracing of the product. The extensive traceability uses the data more broadly. Besides using the data for the product history and timeline, the product tracking data is exploited for managing, optimizing and controlling the process. This allows multiple extensive use cases for traceability data. Example of these is quality variation detection from the process steps which helps in continuous improvement. Because extensive traceability exploits the data in many different areas, the good linking between tracking data and product data is crucial. (van Dorp 2002)

2.2.3 Traceability Systems

Traceability system is used to link the process data and product data for achieving and managing traceability of items and lots. The design of traceability system consists different elements. Töyrylä (1999) summarized 2 models for traceability and material tracking to identify key elements needed for implementation of traceability systems. These models are Steele's (1995) and Caplan's (1989) models. Steele (1995) focus on 4 elements of traceability needed for traceability system. These elements are physical lot integrity, data collection, lot-process liking and reporting. Should be noted that Steele's model is for lot-level traceability where lot size can vary.

On the other hand, Caplan (1989) has 5 different techniques for material tracking. These techniques provide different data types to track the product life cycle. Techniques proposed are lot integrity control, processing control, build control, inspection and test, and field activity and modification control.

So, the summarization made by Töyrylä (1999) consist 4 points which are physical lot integrity, data collection, product identification and lot process linking, and reporting. Physical lot integrity means defining the batch or lot size to achieve the resolution of needed traceability system. 3 possible reasons for decreased lot integrity can be found. First a lot mismatch where the lot is not strictly matching the source lot. Second reason is lot-end mix in which the clear separation of different lots is not maintained, for example due to rework of an item at the end of the shift. Third, in first-int-first-out (FIFO) production, lot-sequence mix can cause loss to the lot integrity. (Töyrylä 1999)

Data collection element includes methods for collecting lot tracking data and process information data. Then, lot-process linking is referencing physical product data and process data together. The product data can be for example specifications, drawing sheets or BOMs of the material or production item. The link can be done for example using identification numbers in physical items and in process data. Other valid option is to use date and the time in data records. The final element, reporting, is needed to get the data from the traceability system. The design must consist for example data storage design and limits, data access intervals and permissions and data structures. (Töyrylä 1999) Moreover, the Steele's and Caplan's models are focusing on lot level and Töyrylä (1999) proposes that small changes are needed for the item-based traceability and reporting, mostly to the physical lot integrity design, when a lot is replaced by an item. Nowadays, lot-size one traceability is often used when item-based traceability is discussed.

Of course, mentioned elements are not guarantee for successful traceability system, but information quality and usefulness of traceability system data, and technical and organizational setting enablers are needed to construct the valid system. The information provided should be comprehensive, accurate and secure. It also should have a timeline and should be available easily and as real time as possible. Data should also be stored well to obtain and access it also after a long time. (Töyrylä 1999)

The technical enablers are improving the traceability information to make it more useful. For example, ADC with automatic identification of production items and lots can improve the data dramatically when compared to manual data entering. The data is more accurate when human errors are eliminated. The data is also available faster with ADC. The technology itself should not be the target but the real value is the collected data and information formed from it. Then, the usage of data to gain business advantage should be in the centre of the traceability system design. Organizational enablers focus more on data utilization through perceiving opportunities when collected data is useful and available. (Töyrylä 1999)

Töyrylä (1999) notices that different studies argue differently about designing of the traceability systems. Some studies think that bottom-to-up approach is the only valid option for designing the system while some studies argues for top-to-down approach. The main reason for bottom-to-up approach seems to be that the true operations of the company are reflected better in that approach. Top-to-down approach is said to be crucial to success because the top management understanding is important. Nevertheless, the motivation and acceptance of operators and other users is crucial to overall success of the traceability system project.

2.2.4 Modelling of Traceability Systems Information Exchange and Data Models

It is noticed in Khabbazi et al. (2010) paper that some products and manufacturing systems are not suitable for using pre-designed traceability system models which means that new model is needed. User friendly, easy-to-understand but at the same time useful and efficient systems are not easily implemented so modelling and gathering requirements for tracking the material is important. At the same time, real-time monitoring is preferable. Nowadays, there is lot of different ways to collect the data automatically by identification of items and lots, which makes the challenge and target to lie more at the data storing, exchanging and reporting. (Khabbazi et al. 2010)

Designing of information models is critical for successful data exchange at least when the company has sister plants, or the trackable product is also monitored through supply chain (van Dorp 2002; Khabbazi et al. 2010). If there is lot of plants, sites or other companies sending tracking data, the information flow may rise very high (van Dorp 2002) and information models will help to handle the information exchange effectively (Khabbazi et al. 2010).

The information systems of company are described by information models. The models can be utilized in 3 different ways (Khabbazi et al. 2010). First, the model can be used for information system identification. The models can also be used to standardize the data which helps to develop information systems. Third way is to use the models for integration of different functions inside the organization. The information models can work like interface between functions to allow access to different properties (van Dorp 2002). Van Dorp (2002) also points out that the information exchange should not always be two-way but sometimes the one-way property access is enough. Nevertheless, the information models should concentrate to the information quality (van Dorp 2002). The important properties of quality are for example accuracy, response time, precision and source of the information.

To successfully model traceability systems, the traceability level needs to be decided and recognized. The traceability level decision also defines the importance of the items of interest. The 2 common levels of traceability are batch/lot level and item/part level (Khabbazi et al. 2010). Third level in some cases can be type level (Dai et al. 2014). The factors affecting to the traceability level, and through that to the item importance, are value of the item, criticality of the item and the external environment. Of course, also costs of the tracking are affecting to the level. (Khabbazi et al. 2010) As an example, the air plane industry demands very high traceability for every part to allow finding the history for faulty parts and components, because it is critical to know which machines are used to part processing or which CMMs are used to check the quality of the part. In this way, the faulty process steps can be noticed to avoid faulty parts in the future. Therefore, the industry type affects a lot to the traceability level decision. Sometimes it is enough to know the history in a lot level, for example if the production batches are made from material from different suppliers. The experience from the big Finnish FMS manufacturer tells that some manufacturers wants to track the production in the item level, but the items are also identified in a batch level meaning that the specific production item has multiple different codes, for example a serial number and a lot number.

Van Dorp (2002) presents some possible benefits of good product identification. These benefits can be used as a feature in the traceability system model. The features can be efficient storage information, sorting of products, tracking the WIP, and tracking and managing the transportations. These allows better view of manufacturing process and business.

One important aspect of traceability models is a data model which is visual plan for database building. The data entities and their relationships are presented in a diagram. 4 different kind of methods for creating the data model are identified and they are presented in the table 11. (Khabbazi et al. 2010) The data modelling methods varies and each of them have a use in different cases.

Table 11. Data modeling methods (Khabbazi et al. 2010)

Method	Description	Features
Richard Parker	A style of visual language to draw ERDs.	Readability and efficient to use for drawing space and data model.
Integration Definition for Information Modeling (IDEFIX)	FIPS based the IISS. Supports management of the data as a resource, the integration of information systems, and building of databases	Used to produce a graphical information model. Basic constructs: Box (to keep the objects), Lines (to connect the boxes), Attribute names (to describe the characteristics of object within boxes)
Entity Relationship Modeling (ERM)	Relational schema database modeling. Uses ERDs.	Different views of data: the network model, the relational model, and the entity set model.
Unified Modeling Language (UML)	A graphical language for specifying, visualizing, and documenting the objects.	3 different views of a system model: functional view, static structural view, dynamic behavior view.

For creating the data model for traceability system, both conceptual and physical modelling is needed. The conceptual model includes steps like identification the traceability requirements, identification of entities and their relationships and integrating the final conceptual model for traceability system. Moreover, the physical model needs identification of keys and storage requirements and construction the physical model for the traceability system. The physical model for example defines the database tables and relationships between tables. (Khabbazi et al. 2010) The table 12 presents simple example of identifying entities for conceptual data model.

Table 12. Entities for the conceptual model based on Khabbazi et al. (2010)

Entity	Description
Order	Order details, including for example ordered amount, customer and due date.
Production plan	Detailed production plan specification
Operation	Information of production operations
Lot	Unique lot (batch or item) in the system
Relation	Information of where-used and where-from for lots

The entities identified in the table 12 can then be used for creating traceability model presented in the figure 32. It should be noted that in the real system model, this step includes requirements planning.

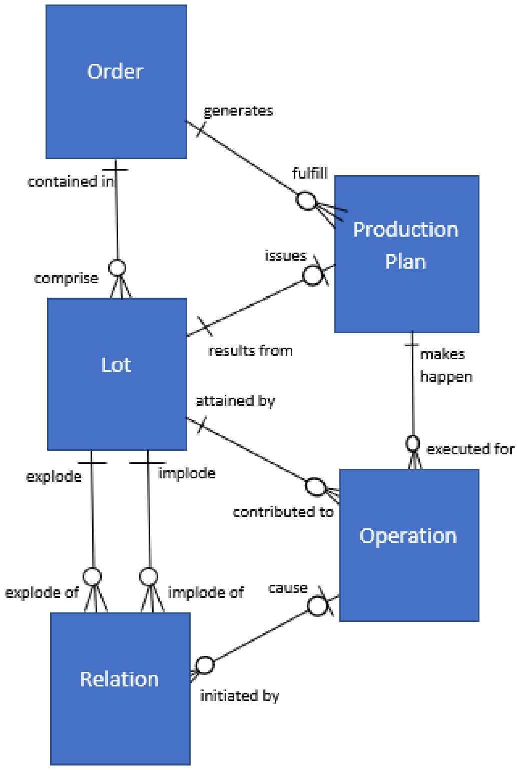


Figure 32. *Traceability model for the conceptual model, based on Khabbazi et al. (2010)*

From the traceability model different relationships can be identified. For example, it is possible to track which operation belongs to which order. This is possible by linking production plan entity with the operation entity. This is possible if production plan executes one or more operations. Another example is relation between lot and order entities. Order entity consists zero or more lots. Also, every lot should belong to some, but only to one order.

The implementation of physical model for current example starts by identifying primary and foreign keys of database tables for all entities. The keys are presented in the table 13. Each entity is presented by a table at the database. The entity attributes are used as database columns. For example, the lot entity forms a database table which structure is presented in the figure 33.

Table 13. *Primary and foreign keys for entities for the physical model based on Khabbazi et al. (2010)*

Entity	Primary Key	Foreign Key
Order	OrderSerialNbr	
Production plan	ProductionPlanId	OrderSerialNbr
Operation	OperationId	ProductionPlanId, LotSerialNbr
Lot	LotSerialNbr	OrderSerialNbr, Production-PlanId
Relation	RelationId	OperationId, LotSerialNbr

Lot		
LotSerialNbr	IDType	PK
OrderSerialNbr	IDType	FK
ProductionPlanId	IDType	FK
Description	string	
Value	interger	

Figure 33. *Entity Lot presented in physical model, based on Khabbazi et al. (2010)*

In the figure 33 physical model of lot entity presents the database structure and data types in the database based on table 13. IDType is unique identifier, and PK stands for primary key and FK for foreign key. When same types of database structures are developed for every entity, physical model of traceability system can be created. The example is presented in the figure 34.

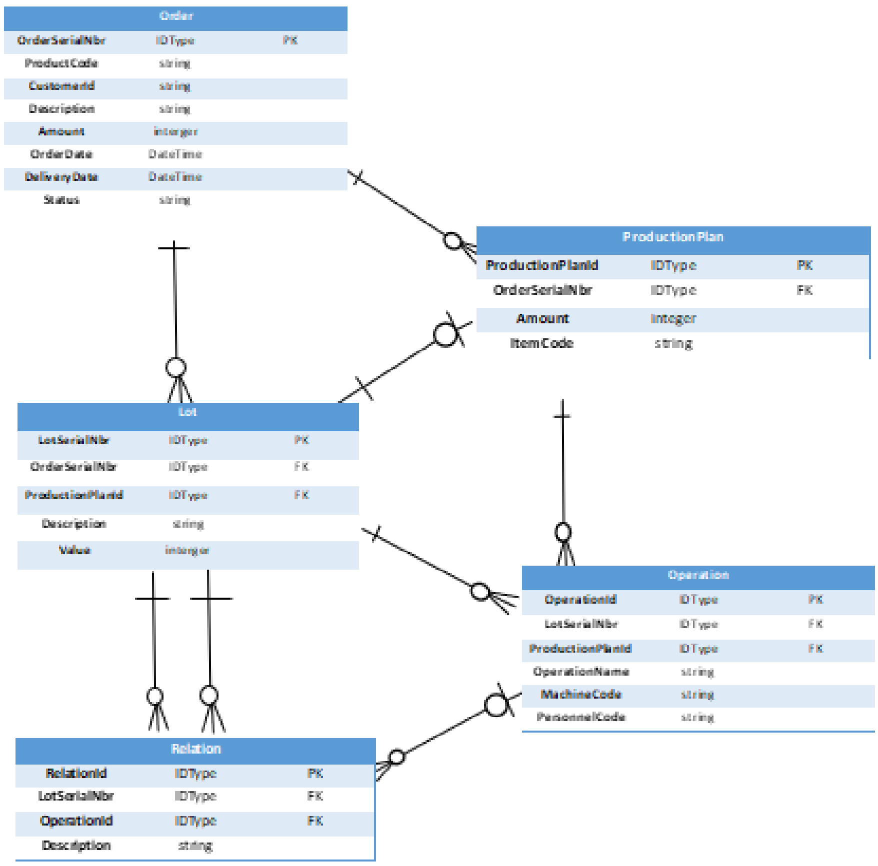


Figure 34. *Physical model of traceability system is forming database tables and their relations, based on Khabbazi et al. (2010)*

As can be seen, the conceptional model and physical model are looking lot of each other. The physical model also describes the relationships between database tables. For example, one lot is attained by zero or more operations and one operation causes zero or more relations. Again, one lot implodes or explodes zero or more relations.

The up-to-date data model in traceability model is critical for reporting by making the data available for computational interfaces. The data is also more visible. The good model also makes the transaction from the model to implementation of the real system smoother and faster and helps the designers and developers to understand the system. (Khabbazi et al. 2010)

2.2.5 Monitoring Supply Networks

Even though the outline of this thesis is focusing more on enterprise level traceability of products, the supply network monitoring should be discussed when full product history is wanted, at least in the current globalization trend in industry where a network of different companies is affecting to the product time line all over the world (Dai et al. 2014). When the whole supply chain, or even the external stakeholders, is tracked, it is even more critical to design the traceability model carefully to integrate supply chain effectively and to gain the needed data without too much data overflow (Khabbazi et al 2010). The large supply chain is also making the traceability more vulnerable to the faults (Dai et al. 2014). It should be noted that some products must be monitored through the supply chain to make the recall of the product possible. This kind of products exists for example in the food industry where a bad material must be tracked back to the source (van Dorp 2002; Dai et al. 2014). However, inadequate identification of products through supply chain causes also good products recalls which is very high financial loss (Dai et al. 2014).

The integration of companies in supply chain need to happen at physical, information and control layers. This needs lot of design and implementation of different concepts. There are ready SCM and ECR applications which can be used for helping the integration (van Dorp 2002). The integration and material flow in supply chain are described in the figure 35. Dai et al. (2014) reminds that usually resources for supply chain traceability system are limited so designing integration cost effectively ways is important.

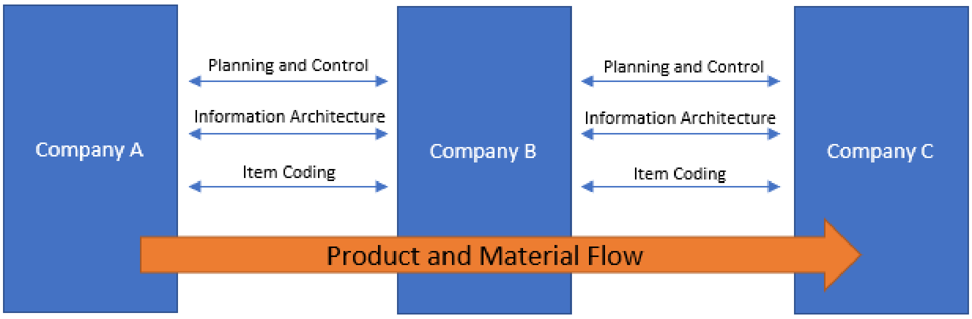


Figure 35. *Integration of companies in the supply chain for product traceability based on van Dorp (2002)*

In the figure 35, the item coding stands for identification of the product using some identification method. In turn, information architecture is made for information exchange and data certifications. Planning and control can be used for recipe optimization for example. (van Dorp 2002) As can be seen, the material flow is from Company A to Company C, but the information and data is going to both directions.

Sometimes it is not easy to persuade every parties of the supply chain to join the traceability system. Then proper incentive method may help to encourage all parties to the project. There are multiple different incentives like cost sharing in case of possible recalls. (Dai et al. 2014) The supply chain traceability also creates visibility through the supply chain (Töyrylä 1999) which might interest also other parties inside supply chain. This might, for example, help the purchaser to plan the distribution better (Töyrylä 1999).

One important use case for the supply chain flow monitoring can occur when the manufacturer has multiple suppliers of the same material, for example to avoid supply risk (Dai et al. 2014). The figure 36 illustrates the situation where the manufacturer is in the first layer and two same kind of suppliers in the second layer.

Figure 36. *Supply chain with two identical suppliers and one manufacturer*

In this situation, suppliers provide identical material batches to the manufacturer company which processes the batches and then send them to customers. When there is a fault in the material, it is important for manufacturer to know who supplied it. Also, the timestamp of the batch helps to define if the other ready product can contain faulty parts. (Dai et al. 2014) Moreover, effectiveness, quality and speed of the supplier can be calculated based on flow monitoring results.

Also, different kind of logistic and security reasons speaks for the need of the supply chain traceability. If the products are marked in and out at some node in every company, the logistic and product shipping time can be tracked. Töyrylä (1999) also states that in the Silicon Valley, lot of components are stolen every week. This illegal activity can be prevented if components and their shipments would be tracked.

One challenge for supply chain traceability is to make the data exchange work in real time and to collect and report it fast and easily. Too difficult access to the data will reduce the usability of it. There would lie the possibilities for Inspector flow monitoring system because it can work as a cloud service and collect data from multiple plants and combine the data to the same database.

Because usually companies and organizations do not operate without network of other companies, also traceability systems could focus on relations between these sister companies, suppliers and customers. The new requirements for tracking caused by new relations are forming traceability system business scope of 4 perspectives which are presented in the figure 37. The perspectives are the enterprise perspective, the multi-site perspective the supply-chain perspective and the external environment perspective. (van Dorp 2002)

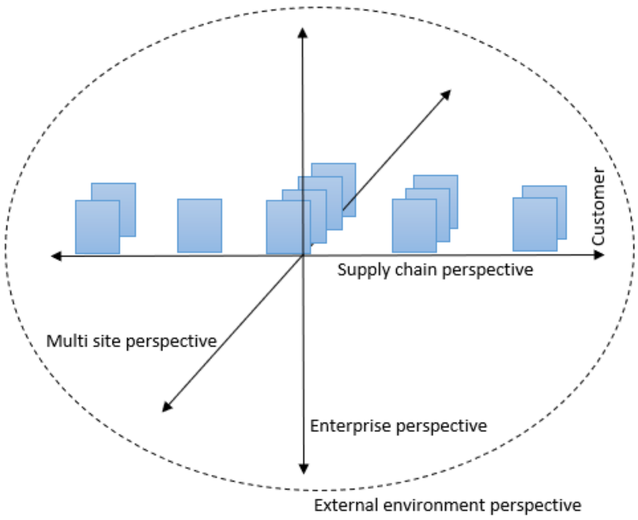


Figure 37. *Four generic perspectives of business scope of traceability system based on van Dorp (2002)*

The tracking happening inside the company belongs to the enterprise perspective. The tracking data may be generated all over the company plant. The traceability in the company can happen only in horizontal way meaning keeping of history records of products, but also vertical approach is possible. In the vertical way the information of different management levels is adopted to the traceability data which allows for example controlling and operational planning. (van Dorp 2002)

The multi-site perspective may be relevant for companies which have multiple plants producing items in different sites all over the country or even the world. The processing may be started at other site and finished in the other site, and the traceability of product needs to happen at every sister plant. At some point, the data collected in different plants should be exchanged to get the full timeline for the product. (van Dorp 2002)

The supply chain perspective means focusing on product movements from the supplier, through the manufacturing and distributing to the end users. Of course, when there are multiple companies involved, there is lot of challenges. For example, the different requirements and data exchanging may become a challenge. The supply chain perspective has both business-to-business and business-to-consumers requirements. (van Dorp 2002)

Business-to-administration requirements are caused by authorities and laws. The external environment perspective is focusing on these requirements. For example, European Economic Community (EEC) has directives affecting to the traceability of organizations. Example of the directives are ‘packaging and packaging of waste’ (94/62/EEC) and ‘the official control of foodstuffs’ (89/397/EEC). (van Dorp 2002)

2.2.6 Optimization the Material Flows with Production Equipment Layout

The plant layout affects to the management of the work but also to the material flows. The demands of the industry and through that the needs for the layout may change from the time when the layout is firstly designed and implemented (Falcone et al. 2014). Proper layout design and workstations locations helps to improve the process performance (De Carlo et al. 2013). By designing the layout that fits the company production, the material flow is improved and the production lead time and WIP part amount is reduced (De Carlo et. al 2013; Falcone et al. 2014). Also, material, equipment and operator movements can be optimized by good layout design (Falcone et al. 2014). Moreover, the layout design affects to the inventory costs, transportation times, delivery times and even to the product quality (Herrmann et al. 1995).

There is lot of different tools for defining the best possible layout. Example tools are path charts, flow process charts, product-quantity data sheets, from-to analysis and relationship charts of service and production activities (Falcone et al. 2014). Common thing for most of the mentioned tools is that by monitoring the material flows in the current manufacturing layout and by valuing distances and quantities, the design of suitable production layout can be started (Falcone et al. 2014).

Of course, not every KPI value can be improved by the selected layout. That makes it important to choose the goals of the layout change (Falcone et al. 2014). Possible targets can be for example material flow linearization, reduction of material exchange distance or reduction of WIP and warehouses. For every target, a suitable KPI is selected and implemented so that success of the layout change can be monitored (Falcone et al 2014).

2.2.6.1 Typical Manufacturing Layouts

4 different types of typical layouts can be defined. The use of the type is relying on production quantity and production variety. The table 14 introduces the types.

Table 14. *Typical production layout types based on De Carlo et al. (2013)*

Type	Description
Fixed position	For production of very big products with low quantity. Example are ships and aircraft.
Job-shop	Production is divided into the technology specialized departments. Flexible.
Cellular	Production is divided to manufacturing cells which are producing a same kind of product or few different kinds of products. Production quantity is usually batch production.
Flow line	Used for mass production of one product and large quantity. Usually highly automated.

Table presents the possible layouts only roughly and potential layout configuration amount is much higher. The 4 typical types just help to detect the usually used types based on production quantity and number of products. The figure 38 presents the layout types in production quantity – production variety diagram.

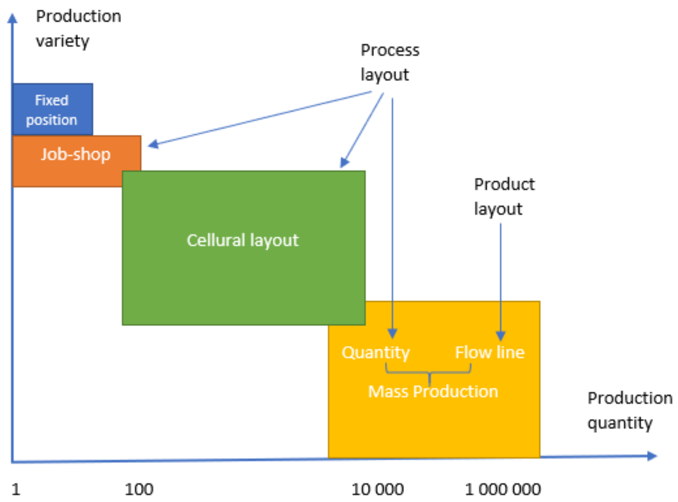


Figure 38. *Variety-quantity diagram of the layout types based on De Carlo et al. (2013)*

The layout types are divided to 2 main types which are process and product layouts (De Carlo et al. 2013; Falcone et al. 2014). In the typical process layout, activities are grouped by the action they perform, while in the product layout, the activities are in a line. The product and process layouts are presented in the figure 39.

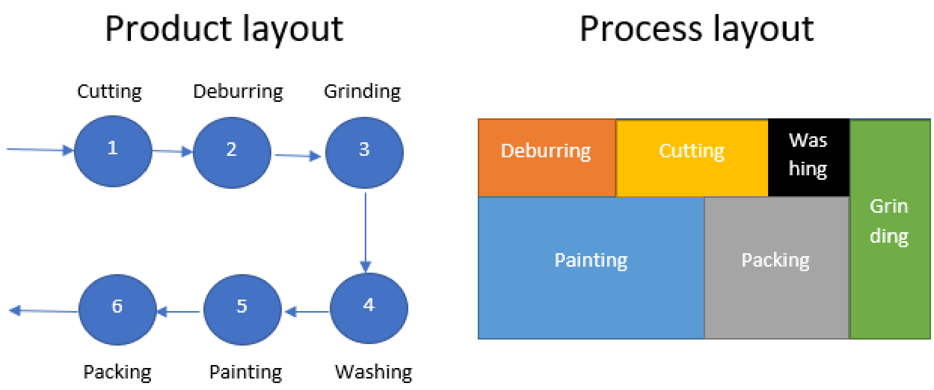


Figure 39. *Production and process layouts*

The figure illustrates how to manufacture layouts are differing. In the product layout, the material flow goes through the activities in sequence. This is used usually in mass production. The process layout needs more WIP buffers and storages because usually products are transferred to the next activity in batches. However, the process layout is offering more flexibility in case of multiple different products because different products might be following the different manufacturing routes creating multiple different material flow paths. Nevertheless, there is also different flexible layout configurations like distributed layout, modular layout, reconfigurable layout, agile layout and U-SPAH layout (Falcone et al. 2014).

2.2.6.2 The Material Flow Analyzing and Improving as Part of Layout Design

By analyzing the material flow, lots of data from current layout can be achieved and used for help to a new layout design. The material flows and relations between different workstations will help to decide the best possible layout for material flow optimizations, operator movements or any other targets and goals of the new layout. Therefore, the material flow path design is very important part of successful layout design (Herrmann et al. 1995).

The systematic layout planning (SLP) is widely used method for designing layouts of the factories. SLP consist 3 phases which are data collection and analysis, figuring out possible layouts and then choosing the best one of them. In the first step, the material flow between workstations is collected and analyzed. The data is then formed as relationship chart which is visualizing workstation relationships, requirements and reasons for relationship. The next step is to create the relationship diagram which is connecting the activities with the lines. The desired closeness of the activities is then defined by the number of connection lines. After the relationship chart and relationship diagram are formed, the suitable layout configurations are searched and decided. The decision should be a combination of economic reasons and other wanted improvements like making the material flow better or decrease the WIP buffers or waste. (De Carlo et al. 2013)

The type of material handling system often affects to the material flow path and to the manufacturing layout by the way the system can control the material flows. Herrmann et al. (1995) suggest that combination of good material flow path and control schema should be found and constructed to get the best possible layout.

2.3 Material Flow Data Collection, Data Analyse and Data Visualizing

To monitor material flow, the item or batches should be trackable so that their location at certain time can be identified. This forms lot of data which should be handled and stored. The data needs analysing to be useful and visualising it will make it understandable. This chapter introduces few data collecting technologies and data storing and visualizing methods.

Generally, the tracking system based on automatic identification of products can be thought as the framework of layers. Oner et al. (2016) designs framework with 5 cross-sectional layers while Wang (2014) presents the framework with 6 layers. The frameworks are quite similar but the system design of Wang (2014) has also control and decision making involved which makes the system so called II-RFID system. Common layers for both approaches are physical (Oner et al. 2016) or asset (Wang 2014) layer, data capturing front end and data capturing layers (Oner et al. 2016) or data acquisition layer (Wang 2014), and processing modules layer (Oner et al. 2016) or control and database layers (Wang 2014). Finally, the top layer for both approaches is layer containing ERP systems and production management. Oner et al. (2016) call the layer application layer while Wang (2014) calls it management layer. Wang (2014) also determines decision support layer between database and management layers. The figure 40 presents the layers based on Oner et al. (2016).

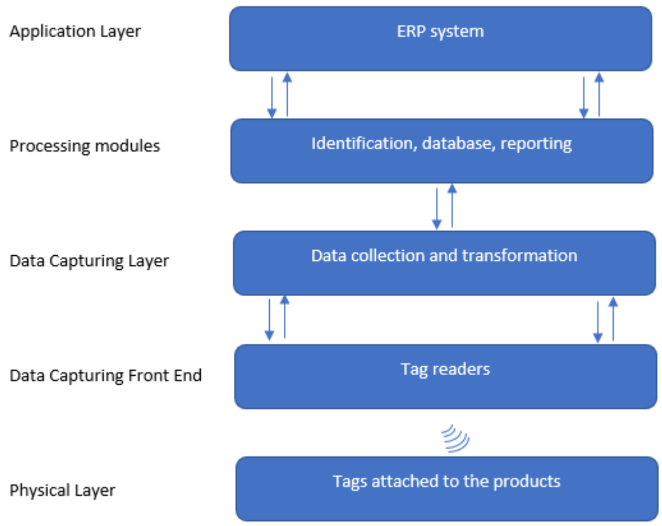


Figure 40. *Architectural framework of automatic identification tracking system based on Oner et al. (2016)*

The physical layer is tags, for example RFID tags, attached to the material and products. The readers or other tag detecting devices belongs to the data capturing front end. Usually, the devices are located at specific points which are called nodes at this thesis. This allows real time data capture from the system. The data capturing layer is converting data to the business data and distributes the clean and accurate data to the processing modules. The data capturing layer also manages the reader devices and tags. (Oner et al. 2016)

The events and reports are generated by processing modules layer which is requesting the data from the data capturing layer. The data is stored to the databases and processing module layer handles it. The fifth layer, application layer, takes care of core business processes. Usually this layer includes ERP system and different kind of WEB applications which are supporting the business. This layer also works as user interface to the application. (Oner et al. 2016)

2.3.1 Data Collecting Technologies in Traceability Systems

To make traceability of production possible, the items or batches must have mark which makes identifying them possible. There is lot of different methods for detecting the item but two main classes for marks can be found. First, the permanent marks are moulded, etched with laser or casted to the items. These marks are permanent and cannot be easily removed. The second class of marks contains different kind of attachable marks like barcodes and radio frequency identification (RFID) tags. This kind of automatic identification methods also includes technologies like magnetic inks, voice recognition, smart cards and biometrics. (Wang 2014)

2.3.1.1 Barcodes

Almost every item which is bought from the store has barcode attached to it. Usually barcodes are divided to two main groups which are 1D and 2D barcodes. The difference is that the 1D barcode holds data only one-dimensional but 2D barcode holds data both horizontally and vertically. 2D barcode is easier to expand because the barcode can be expanded to both directions allowing more data and still easy scanning. (White et al. 2007)

Because the barcodes are visual, they are not affected by electromagnetic emissions or other radiations. Barcodes can also be printed on materials which are very durable. The disadvantages of the barcodes are that they need line of sight and therefore the dirty barcodes cannot be read. This might be very disturbing in production environment. Also, after printing the barcode it cannot be updated anymore. (White et al. 2007)

Using the barcodes in the single-lot traceability is not possible in all the systems because barcodes can only identify type of item instead of unique item. Also, often in the production environment, barcode systems are causing inconsistent data and therefore reports are unreliable (Oner et al. 2016).

2.3.1.2 RFIDs

The RFID system can be thought as transceiver-transponder system. The RFID tag carrying the data is the transponder while the RFID antenna is the transceiver. There is 2 different kind of tags typically used. The internal battery of RFID tag is used as a power supply when the active tag is transmitting signal to the antenna continuously. In turn, the passive tag receives a low power signal sent by the antenna and uses it as a power for data exchange. Despite the differences, presented in the table 15, of active and passive tags, the carried data is the same and it is called an electronic product code (EPC). EPC is used for assigning all the items with a unique identification making it possible to trace the item. (White et al. 2007)

Table 15. *Difference between active and passive RFID tags based on White et al. 2007)*

Active RFID tags	Passive RFID tags
Internal battery powered, finite life time	No power source, powered by signal from RFID reader
Long range	Short range
Not so sensitive to interferences	Sensitive to interferences
High data transmission rate	Low data transmission rate
Possible to read multiple tags simultaneously	Only few tags can be read simultaneously
Reader does not need to be aimed precisely to the tag	Reader must be aimed to the tag

Despite the differences between the active and passive RFID tags, the hardware of the RFID system is quite identical. When selecting the RFID tag for the monitoring system, the different properties of the needed tag must be well designed. For example, reading distance may vary a lot and it might have huge impact for the system. Other requirements for tag selection could be for example size, mounting technique and material combability with the products. The material combability means that some tags are not usable with plastic products while some tags are not usable with metallic ones. (Wang 2014)

RFID antennas are used for tag detection and they are attached to specific points to detect the item movements. There is multiple different kind of antennas and the type is decided based on the system needs. Additionally, the RFID reader powers up the antennae and sends the tag data forward. Both fixed and handheld readers are used in monitoring. The handheld readers integrate the antenna, the reader and the power supply into one device. (Wang 2014) Often, the readers could also be used to write the RFID tags. Finally, in the RFID systems, the middleware implemented by software or with specific hardware is used to process the RFID data and send it to the database (Wang 2014).

When comparing RFID technique to the barcode technique in traceability, different aspects can be detected. The RFID chip can, for example, get interference if it is used nearby other electronics while the barcode is not affected. On the other hand, RFID technique is faster and more consistent than the barcode system. This creates benefits for accurate monitoring. Also, the RFID tag does not need line of sight with the reader while dirty or badly placed barcode can cause errors while reading. (White et al. 2007) The main differences of the RFID and the barcode are presented in the table 16.

Table 16. *Main differences between RFID and barcode in flow monitoring*

RFID	Barcode
No line of sight needed for reading	Line of sight needed for reading
Multiple tags can be read at the same time	Only one tag can be read at the same time
Is not affected by dirt	Reading dirty or damaged tag is not possible
Unique item can be identified	Only type of item can be identified
Tag can be written multiple times with different information	Update of data is not possible
Data can be encrypted	Easily reproduced which makes them less secure

Like can be seen from the table 16, RFID is often fitting the production environment very good. Therefore, it is not surprising that trend of RFID technology is rising (Oner et al. 2016). To summarize, both technologies are usable in traceability systems and both have advantages and disadvantages. If simplified, RFID tags should be used when data rewriting is important and when single items are tracked in so called single-lot systems. If tracking a batch is enough, barcode is valid option. It is well-known and supported technology which makes adapting easier.

2.3.1.3 Detection Node and Tag Locations

For implementing traceability system, it is very important to design the detection nodes with tag readers for achieving best possible reading performance (Oner et al. 2016) and to collect right kind of data for wanted reports. The detection node is the location where the tag reader and antennae are placed or attached. The amount and location of detection nodes must be designed, and the decision must include the analyses of the wanted reports. For example, if the product machining times and machining devices is under interest, the detection nodes must be placed within the machines. On the other hand, if monitoring of WIP buffers is wanted, the detection nodes must be placed near the buffers. One important design case is if the data is reported when the tag enters, stays or leaves the detection node (Wang 2014). It affects to amount of detection nodes and to the locations of them.

To connect the identification data from the tag to the detection node, the nodes must be identified also. This means that every detection node must have identifier of its own.

Then, when analyzing the data, the history of the product can be tracked through different nodes.

The detection node must be located so that automatic detection of the tag is possible. The reader must be aligned to the right direction and it must be close enough of the material flow to detect the tag. In case of manual readers, the reader must be close to the working area and easily accessible by the operator.

The tag location in product is also critical. Some products cannot have a tag attached so that it leaves a mark to the product. Also, the tag should be easy to remove if there is no intention to leave it to the ready product. Therefore, finding a place for a tag where it can be automatically detected without causing permanent marks to the product is very important. For example, the tag can be attached to the pallet or box where the product is located. In case of denim industry, the tag can be placed to the pocket of the clothes. Of course, there is limitless amount of possible tag locations and they must be designed for every item type.

2.3.2 Is Simple Node-based Material Flow Data Big Data and Where to Store it?

Big Data is collection of huge amounts of data which is varying a lot making it hard it to store to very structured databases, like relational databases (Gašpar & Mabic 2016). Even there might be huge amount of material flow events and therefore vast amount of flow data, the data is usually quite structured and have strong relations to the other data. For example, the data event from the detection node might have relations to the product data, order data and detection node identity data, and all data is very structural and can be stored to firm tables with predefined columns. Therefore, the flow data is hard to see as big data. Also, when comparing to traditional Big Data forerunners like Google, Yahoo and Amazon (Gašpar & Mabic 2016), the material flow data is, at least usually, lower and data has less varying.

Nevertheless, to collect and store material flow data, data storage must be well designed. The data storage must hold huge amount of data, and the data must be stored in understandable format and it must be easily accessed. With the vast amount of data, the data storage speed is also playing role. Usually databases are used as a storage because they are fast, reliable and easy to access by software.

The event data received by tag readers can be stored into relational databases. For example, Wang (2016) uses relational database in his study of RFID tracking system. Also, White et al. (2007) suggest storing data to databases for later access. The events can happen rapidly and therefore a good data storage and storage structure is needed. It is important to design tracking event database and data structure so that it is linkable to the other data tables like production order and product information.

2.3.3 Data Visualization Importance in Enterprise level

Data collecting without processing and reporting it is waste of time and money even if the data is very accurate and relevant. For reporting the data, KPIs are effective tools because they provide much information in simple format and understanding the system performance gets much faster and easier. Other effective way of reporting the data is data visualization. More likely everyone has heard a saying “a picture is worth a thousand words”. That is true also in enterprises and in case of data reporting. A simple graph or chart can give information about system much more than huge amount of data lines and text. For example, different kind of trends might be recognized from the visualization. Also, many times people use to trust and remember visualized things better than read and heard because visualizing makes things concrete (Hepworth 2017).

Data visualization is converting data which is in tabular format to graphical presentations like charts, maps and tables. When designing data visualization, people trait for emotional thinking must be taken in to account. This can be done by using different colours, fonts, icons and other visual aspects. (Hepworth 2017) Many times in enterprise environment, negative data is presented with a red color while positive data is presented with a green color. Then, only by watching the color of visualized data, the message of the data can be known. With further exploring of the visualization, more information, like exact values, can be gained. This so called ‘visual language’ is very complicated and cannot be explained same way than written language, and for example cultural differences affects to the visual understanding – for example in China, red is not a color for danger but for fortune and good luck (Hepworth 2017).

There is research area called InfoVis, information visualization, which focuses on making data analysing and understanding easier for users by visualizing it (Liu et al. 2014). In the enterprise environment, InfoVis supports understanding of data and therefore makes data utilization easier. Right utilization of data can lead to growth and better productivity. (Liu et al. 2014)

InfoVis have pipeline of 5 main modules which are data transformation and analysis, data filtering, data mapping, data rendering and UI controls. Additionally, before the data transformation, there is data collection which is acting as input for a visualization pipeline. The pipeline is demonstrated in the figure 41. (Liu et al. 2014)

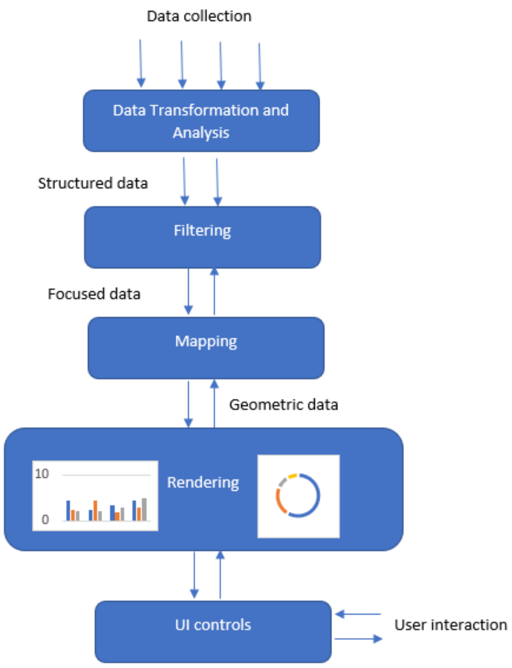


Figure 41. *Visualization pipeline of InfoVis based on Liu et al. (2014)*

The input data can be very unstructured, and the data transformation and analysis module take care of structuring and adopting the data to the visualization needs. Also, simple filtering can be done already at this module, for example removing faulty measuring values. The next module then focusses the structured data and selects data pieces for visualization. After that, the mapping module is attaching visual attributes to the data. These are for example color and size. Also points and lines of visualized data is selected there. The data is then transformed to the image data by the rendering module. Finally, the UI controls modules is used by the users for exploring the visualized data. (Liu et al. 2014)

To implement InfoVis in real world applications, empirical methodologies are used to found suitable designs and implementations. That is reason for why visualization development is made with real industrial data. Also, user interaction is important in InfoVis where multiple categories of user interaction are defined. These are for example reconfiguring, exploring, selecting and filtering. (Liu et al. 2014) This is important in industrial environment which has lot of data from big time range. For user, it might be important to get specific data by filtering time, products or machines. In material flow monitoring, it might be crucial to get the data of the single product for recalls and for the product flow history. Therefore, it is important for user to have system where interaction with the data queries can be done.

The data visualization is not only tables and diagrams but there is limitless amount of visualization methods. Purpose of the data visualization is not the visualization itself, but it must support the data analysis. Therefore, different methods for visualization of data can be developed for different needs. For example, typographical maps, event and theme rivers, matrixes, and many other visualization methods can be used (Liu et al. 2014).

Hepworth (2017) presents 2 modern trends in visualization of huge amount of data. These are interactivity and real-time updating. The interactivity perspective allows multiple spatial configurations to make exploring different levels of data possible. For example, the user may find new visualizations from different perspective by clicking different data points. In turn, the real-time updating data visualization is listening changes in the data set and updates the visualization in real time.

Of course, visualization of data is not simple, at least when complex data is used. Liu et al. (2014) brings out multiple different technical challenges of visualization. The data visualization usability is the first challenge. As written earlier, the visualized data should help the user in data analysis, which is a challenge for the designers of the visualization. The visual scalability is the second challenge. The visualization tools should be able to present huge amount of data and then even the display capability might be limited factor. Multiple different techniques could be used for helping in this issue like filtering, clustering and sampling. Still, the challenge is to provide enough information and details when data amounts are increasing.

Integrated analysis of heterogenous data is the third challenge. Nowadays, the data is varying a lot and it might be collected from multiple sources, which makes it hard for developers to form accurate visualizations. The streaming data is causing the fourth challenge when visualization must be generated incrementally every time new data arrives. For example, physical components like processors and memories may struggle in computing power when data queries are made frequently. (Liu et al. 2014) This is also the challenge in visualization of industrial flow monitoring data where new data entries are generated frequently. Therefore, the data queries and modeling should be designed as effective as possible.

The fifth mentioned challenge of data visualization is error and uncertainty in data entries. The data may contain noise or other faulty information due the sensor faults or human errors and therefore the visualization might not be truthful. The challenge is to effectively indicate the data errors and uncertainty to the user. (Liu et al. 2014)

The visualization platform should also be thought when developing visualization software. Sometimes it is enough that data is accessible from supervisor's office but many times it would be useful to access the data also with handheld devices or with big dashboards. The different web standards and cloud services create good base for visualization allowing access from multiple platforms and locations.

3. EXISTING COMMERCIAL SOLUTIONS

Usually it is better to get knowledge about competitors and third-party applications before using lot of time and money for designing and implementing an own system. This chapter focus on possible commercial solutions on monitoring area. Also, two possible visualization and analysing tools are presented and estimated. Using ready visualization platform would allow concentrating on data collection and analyse instead of visualization designing.

3.1 Possible Commercial Applications

It is not easy to find information about production flow monitoring and traceability applications. However, 2 possibilities are found and presented. These are Plex Manufacturing Cloud and Zebra's software. Zebra has 2 possible options, Zebra Material Flow Suite and Zebra Savanna. Unfortunately, deeper information on both Plex and Savanna is not found but quite good overview of them is still constructed for decision making.

3.1.1 Plex Manufacturing Cloud

The Plex Manufacturing Cloud is software platform marketed to be more than ERP system for discrete manufacturing companies, labelling itself as manufacturing ERP software. It is built on MES platform making it suitable for many manufacturing businesses. Plex was invented 2001 and it started from automotive metal forming, but it was quickly expanded for different discrete manufacturing industries. Nowadays, over 600 organizations are supported with Plex in over 22 countries. Plex is running totally on cloud service which also allows use of SmartPlex on mobile devices to connect to the production data. (Plex 2018)

Because the Plex Manufacturing Cloud is acting like ERP and MES combined, it is very complicated and has huge number of features like accounting, customers and sales, human capitals, quality management and planning and scheduling. (Plex 2018) This excludes Plex from direct rivalries of Inspector. However, Plex MES is having features like inventory and production management which are in area of interest of Inspector.

Plex Inventory Management promises accurate and real time inventory management. With accurate inventory Plex tries to support inventory valuation and transparency to get rid of sitting capital on the floor, to minimize operator errors and to reduce scrap and returns. Plex also have barcode printing and scanning system included which helps to track the production item movements, and therefore have traceability possibilities like recall and quality issues. Plex supports scanning every items of inventory from raw material to ready products to make full timeline of the single item. RFID scanners are also supported by Plex.

The production movement tracking manually can be done with Plex Mobile. With the mobile application the operator can mark the from-node and to-node so that inventory management is kept real time. Additionally, Plex Inventory Management have support for inventory payment and invoice management. (Plex 2018)

Plex also offers real time visualized view of the soft floor. The device and production statuses, including alarms, is read with PLCs. This helps operators to detect and solve errors faster. (Plex 2018) This included to the Plex ability to track all material provides important information for good decisions for production.

Plex have application called IntelliPlex Analytic Application which allows monitoring of production with different KPIs like OEE, scrap rate, inventory turns and machine availability. The KPIs can then be visualized on dashboards which can personalized for different users or user groups. Because of the cloud service nature of the Plex, the data and visualizations are accessible everywhere. Ad-hoc queries are also allowed for getting the wanted data fast without code writing. (Plex 2018)

End-to-End material visibility of the Plex is much what is designed for Inspector, while monitoring machine and production status seems to be like in the current Inspector version. Plex allows tracing items from entry point until it leaves the factory to the customer. Unfortunately, information about visualizations of material flow or KPIs are not provided accurately. Nevertheless, the Plex have dashboard which might provide visualizations also for material flow or inventory levels. But to summarize, the Plex is much more complex system than Inspector which is not designed to replace ERP. Using the Plex only for production monitoring would be waste of lot of features of it.

3.1.2 Zebra

Zebra is a company from USA working in multiple manufacturing areas like retail, healthcare, transportation and logistics. Zebra tries to connect people, assets and data with their products, services, analytics and software. Zebra is found at 1969 and nowadays it has about 7 000 employees in over 50 countries. The sales in 2017 was 3,7 billion dollars with over 10 000 channel partners. Zebra is not only providing data capture software and services but own barcode readers and RFID equipment. Other business areas are for example printing, mobile computing and location solutions for sports. (Zebra 2018)

As can be seen, Zebra is not only focusing in production monitoring. Still, they have Zebra's Visible Value Chain solution which is made for real time data tracking and real time material flow visibility. This application is called Material Flow Suite and Zebra's target is to help with process optimising, smart planning and reaching flexible operations.

The Material Flow Suite have 3 core modules integrated together. These modules are Replenishment, Asset Workflow and Equipment Management. The replenishment process is managed by the Replenishment module. For example, automation of inventory replenishment by making operators to trigger material requests helps to reduce inventory costs. Equipment Management provides visibility to material flow equipment and people locations. The locations of assets can be monitored by Material Flow Asset Workflow module. This module is not only making possible physical tracking of items but also informs if some specific item is delayed or is going on wrong route. The other gained information is items graphical presentations where track data points and meta-data of items can be analysed focusing on specific item type or model. (Zebra 2018)

Real time tracking is made possible in Material Flow Suite by combining barcode and RFID technologies with software. Zebra uses open standards on the software which makes third-party integrations easier. Zebra has also mobile solution for tag detection. The great benefit of Zebra system is that they make complete tracking solutions using their own hardware for barcode, RFIDs and other location solutions. (Zebra 2018)

It is quite hard to compare future Inspector with Zebra's Material Flow Suite. There is not much information from visualizations, KPI creations or other monitoring tools of Material Flow Suite. Of course, Zebra's huge advantage is the hardware knowledge. Nevertheless, Material Flow Suite seems to be more than just monitoring tool because it can for example alert from route mismatches. This means that Material Flow Suite should have knowledge about designed routes of the production items. Based on the information get from the web site of Zebra, the Material Flow Suite is designed more for inventory management and material replenishment than to flow monitoring and KPI creating.

However, Zebra has also other software solution called Zebra Savanna which is made for data collecting, data analysing and data applications. Savanna is collecting data from scanned barcodes or RFIDs but also from devices. The data is then stored and processed to make real time and historic analyses possible. Savanna also offers secure APIs for data applications to provide processed data for decision making. Also, different kind of KPIs and visualizations are provided by Savanna. (Zebra 2018) Again, there is not much information available from Savanna but based on gained facts, the Savanna offers great visibility for devices and operations. Still, visualizations or KPIs for material flow data might not be provided.

3.1.3 Possible Use of Third-Party Application

Even though there is not much detailed information available for Plex or Zebra, both applications seems to be focus more on something else than material flow monitoring. The target of Inspector is to provide KPIs based on material flow and make visualizations for the traceability data using for example directed graphs. Plex is more ERP and MES solution than just monitoring system while Zebra's software is focusing on inventory management with their own hardware and tracking expertise. Still, both Plex and Zebra have very good innovations and features for monitoring software. Plex is working totally on the cloud service which makes it easy to accessible with any device. Then also software updates are easily distributed to many customers. On the other hand, Zebra is wisely using the open standards in their software making integration to the other applications possible. Also, both Plex and Zebra have mobile application for tracking which is good area to target also with the Inspector.

Based on the marketing and tutorial videos, Plex user interface is quite clear but it has so many features that using it seems complicated. Zebra's UIs, at least in Savanna, seems to be modern and good looking but not real judgements can be made with only a few marketing videos.

Therefore, can be argued that developing Inspector to contain material flow monitoring is justified. It feels like there is not simple-enough software focusing on material flow monitoring and creating KPIs based on the data collected from material flow. Also, for monitoring application, clear and informative, but still easy-to-use, UI should be designed. Moreover, it seems like adapting Plex and Zebra to ready process might need bigger work than designed adaption work of Inspector. For monitoring system, it should be easy to add to ready automation system without major commissioning breaks.

3.2 Possible Visualization Platforms

One possibility is to collect the data with Inspector but to report and visualize it with the third-party platform. There are many valid options in the market, for example widely used business intelligence applications IBM Cognos BI, QlikView and Microsoft Power BI. This chapter presents 2 commercial platforms which allows integration with collected data. The reports and visualizations are then generated by the third-party platform. One of the options is major player in the market, Microsoft Power BI. The other one is from smaller and more unknown company. It is called PingFlow and their solution is called PingView.

3.2.1 PingView

PingFlow is company located in the northern France and it is expertizing in digital visual management. They have dynamic wallboard solution called PingView. The idea behind the PingView is that external data, for example in ERP databases, web services, Excel files or PDFs, is connected to PingView in real time for creating reports and visualization. Also, cross-database queries from different data sources is possible. The data visualizations can then be displayed with different devices through web browser. (PingFlow 2018)

The user interface is designed to be easily used and wallboard or dashboard creation can be done simple by dragging and dropping widgets. There is lot of pre-designed widgets like images, graphics, maps and texts which can be used for data presenting. User can create multiple dashboard and set them in present mode so that the dashboard shown changes automatically. (PingFlow 2018) The architecture of PingView is presented in the figure 42.

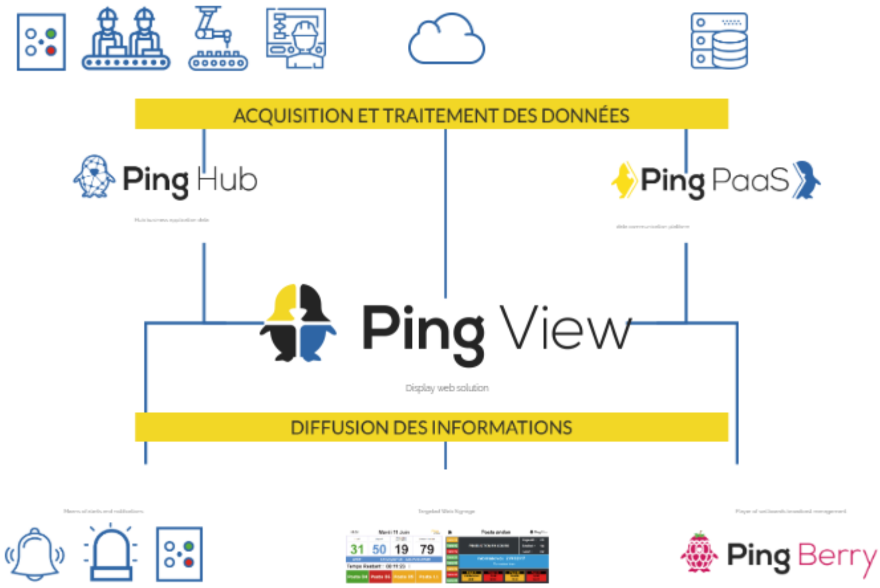


Figure 42. *Architecture of PingView system (PingFlow 2018)*

The data is first collected from the external data sources. PingView makes data acquisition through a data communication platform called PingPaaS which is platform as a service (PaaS) designed to replace the VPN connection. With PingPaaS, the data can be transferred securely between local hosted data platform and the cloud solution. Then data is processed, filtered and formulated. After that, the PingView publish the data to dashboards and creates possible alerts and notifications. The real time synchronization with the data source and therefore the reporting and visualizing is also possible. PingHub is used for operators or other business applications to access the data from PingView. PingBerry is for wallpapers broadcast management but unfortunately there is not much information about it. PingView can work locally or as software as a service (SaaS). The figure 43 gives good understanding of PingView UI and the dashboard creation possibilities. (PingFlow 2018)

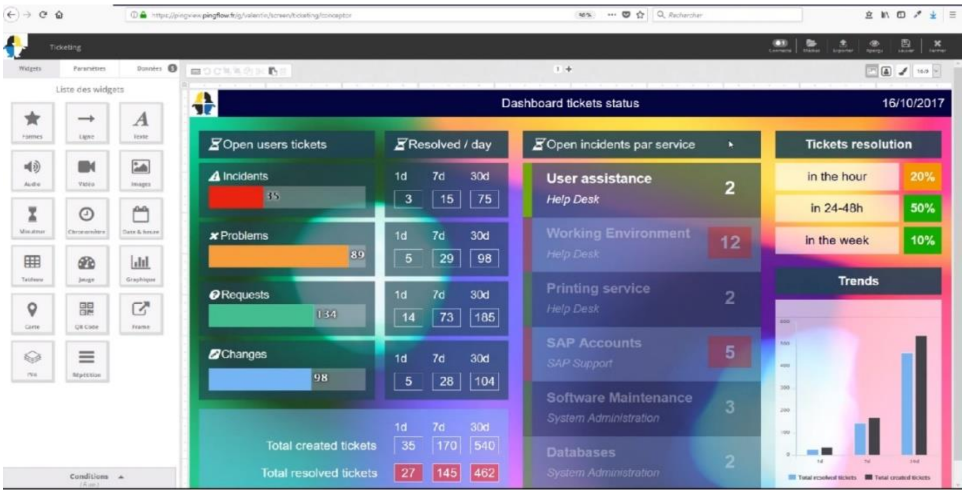


Figure 43. *Screenshot of PingView, captured from PingFlow demo material (PingFlow 2018)*

Even though the visualizations and possibilities of PingView looks good, adopting it to Inspector is not recommended. Inspector stores the material flow data in SQL database which could be ideal for the PingView synchronization but the PingFlow website is not providing enough information about the software. Additionally, the website does not raise trust because there are many links which are pointing to the missing pages. Also, almost every demo video of PingView is in French, which makes to wonder if the target distribution area is only French speaking countries. The same is with the store of the website where French is the only available language. However, PingFlow have some big clients like Scania, Toyota, and Renault (PingFlow 2018) which might tell that the PingView is well working software. Finally, there is not any mention about possibilities to develop own widgets for dashboard which would be very nice option for Inspector point of view.

3.2.2 Microsoft Power BI

Power BI from Microsoft is suite of multiple tools for business analytics. For data querying, Power BI can use multiple different sources like databases, cloud services, Excel sheets, Azure SQL databases, Google Analytics or IoT devices sending continuous data. Power BI website acknowledges that there are hundreds of different possible data sources. The data can be used to create reports and to share them with the organization or supply chain. Also, different kind of dashboards can be created with data visualizations from multiple data sources at the same time. One of the huge advantages is the similarity with Excel which is widely used in the offices around the world. For example, knowledge about Power Queries from Excel can be utilized with the Power BI. (Power BI 2018)

Interactive dashboards and reports can be designed personally. Power BI has over 85 ready data visualization components which can be drag and dropped to the canvas. This allows visualization of many important manufacturing KPIs, like machine utilization, inventory levels and cycle times. Power BI has also open source custom visuals framework which allows creating an own widget if none of existing ones are suitable for planned usage. Moreover, the reports can be created for mobile, and Power BI reports can be published to the cloud for mobile-accessibility. (Power BI 2018)

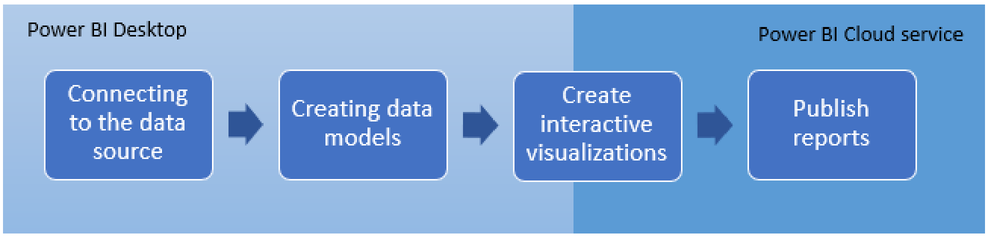


Figure 44. *Basic workflow with Power BI*

The figure 44 illustrates the basic workflow of Power BI. First, the data source is connected to Power BI Desktop application. Then, data models are designed and shaped. This means for example combining different data tables together using identifiers in each table. Power Queries can be utilized for data modeling. After data models are ready, visualizations for KPIs and other data can be made with Power BI Desktop using many different tools. The visualizations and reports can be published to the other people with the cloud service. There is also possibility to embed the reports to be part of own software. This option is called Power BI Embedded and it is introduced later in the chapter 3.2.2.1. (Power BI 2018)

The Power BI Desktop is application used for connecting to data sources and to create visualizations. It is free to download and use. For publishing the data, there is different paid Power BI service options. These are Power BI Free, Power BI Pro, Power BI Premium and Power Bi Embedded. (Power BI 2018)

The Power BI Free can publish data to the Power BI service and embed it in the public websites, but commercial use is not allowed. Data cannot be shared to the users with Free license and data cannot published inside the organization. The users with Power BI Pro license can publish data visualizations to the cloud and access the data later. With Pro license, the data can be published inside the organization. Govern security, like user authentications, is also handled with Power BI Pro accounts. Power BI Pro can also be integrated to other Microsoft solutions like Office 365 and SharePoint. The Power BI Pro means buying a user license for every user. At the end of year 2018, the price for one user license is $9,99 dollars per month. (Power BI 2018)

The Power BI Premium is designed for large volumes of data promising consistent performance. It is more like enterprise BI platform which can be used all around the organization. This means that additional user licenses for everyone exploring the data is not needed. Also, external users can access the data if wanted. For sensitive data, there is Power BI Report Server which allows storing data behind organization own firewalls. This allows using some data from the cloud and the sensitive data on-premises. The Power BI Premium allocates capacity for organization instead sharing common capacity to make sure that Power BI is running smoothly and to allow users to access the data without licenses. That is why the Power BI Premium uses capacity pricing and purchasing enough of capacity nodes for Power BI must be planned. Nevertheless, also Power BI Pro licenses are needed for creating and publishing BI content. (Power BI 2018)

Just to get basic idea of pricing, I calculated Power BI Premium price for company of 50 people with 10 Pro licenses and 30 active users using calculator available in the Power BI website. The 10 employees are counted as occasional users. The price for the Pro licenses is $100 per month and for the Power BI Premium $4995 per month. This is about $100 dollars per month per employees. If the company size is 5000 employees, there are 1000 Pro licenses and 3000 active users, the combined price is $29970 which is about $6 dollars per month per employee. It can be said that Power BI Premium is developed for the big companies to keep the price per user more realistic.

3.2.2.1 Power BI Embedded

From InSolution and Inspector point of view, the Power BI Embedded seems the most suitable version. Power BI Embedded is PaaS which allows independent software vendors (ISVs) to embed Power BI visualizations to own software using interfaces of Power BI Embedded. There are 2 different versions for Power BI Embedded which are embedding for organization and embedding for customer. (Power BI 2018) Because embedding for customer is closer of interest of InSolution, this chapter focus on it. Embedding for customer means that the user must not have the account for Power BI or any knowledge about it (Microsoft docs 2018).

When visualizations are done with Power BI, the developing resources can be used to improve the base application instead of developing own visualizations. Overall, the basic functions of Power BI are used as with other Power BI versions. The visualizations are created with the Power BI Desktop (Power BI 2018). Then the visualizations are published to the Azure cloud service (Microsoft docs 2018). Azure cloud is promised to have advanced data services with reliable infrastructure. Additionally, it should be secure, flexible and scalable. (Power BI 2018).

Then, the visualizations and dashboards in the service can be accessed by own software using the Power BI APIs (Microsoft docs 2018). There are multiple APIs available, for example REST API, .NET API and JavaScript API. The Power BI Embedded also offers own SDK for ISVs. The reports from Power BI are provided with HTML5. (Power BI 2018). The benefit of REST API is that ISV can use any coding language available (Microsoft docs 2018). The basic workflow with Power BI Embedded is shown in the figure 45.

Figure 45. *Workflow with Power BI Embedded*

First, the data visualizations are done with the Power BI Desktop. The data sources are added and modeled in this step. Then, the open source tools, like JavaScript SDK, are used for embedding the Power BI visualizations which are published to Azure. In the last step, the ISV application is registered in the Azure and deployed to the production environment. (Power BI 2018) The figure 46 presents the architecture when Power BI is embedded to own software which is working on browser.

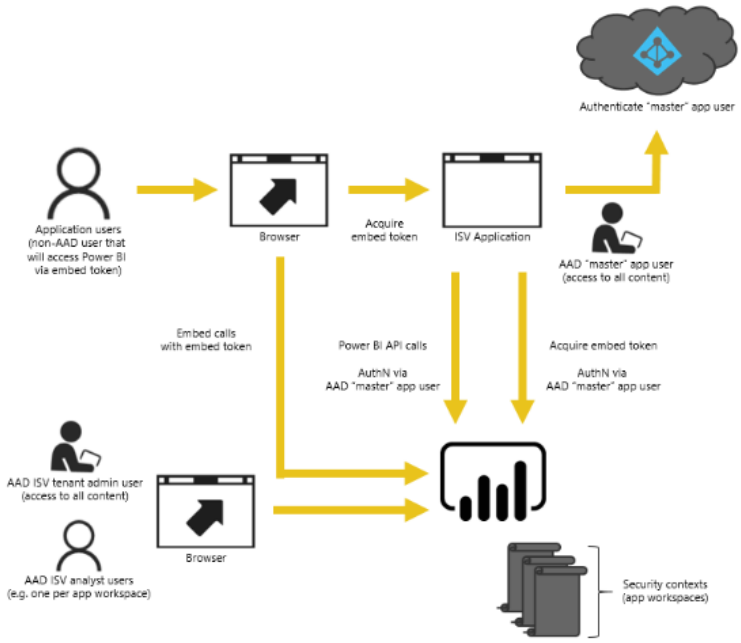


Figure 46. *Power BI Embedded with ISV application adapted from Power BI (2018)*

The visualization data is stored on the Azure, from where ISV application acquires embed token for application user. Then, the application user will get the visualizations visible to the browser. ISV admin user can access and modify visualization through browser. Admin user must also configure master user for ISV application to allow application to get the embed token. (Power BI 2018)

The pricing of the Power BI Embedded happens in capacity-based and hourly measured model, working like pay-as-you-go. Power BI Pro license is also needed for every developer for administrating, content publishing and deploying the final solution. (Ezrachi 2017; Power BI 2018) The needed capacity can be calculated if parameters like user data models, complexity of queries and usage of the application are known. Of course, knowing the exact refreshing rates or usage models are hard to know. (Ezrachi 2017)

The capacity is sold with 6 different groups. These groups have different number of virtual cores and memory, and of course the price is differing between groups. For example, the first group have one virtual core with 3 GB of memory. Then, the page renders at the peak hour can be 300 times. The price is $1,008 per hour. The most powerful version has 32 virtual cores with 100 GB of memory. With this, there can be 9600 renders at the peak hour. The price is about $32 per hour. (Ezrachi 2017)

To calculate the needed capacity, developer must predict the range of page renders at the peak hour of application usage. In the Power BI Embedded, the page render is thought as an action where visuals are loaded from Power BI server. The action can be page refresh or interactions, like filtering, of the visualization. (Ezrachi 2017)

To get some understanding of the pricing, example is provided. If there are 50 users in the peak hour and every of them makes five renders per hour, it is 250 renders per hour in total. Then, the smallest version of capacity option can be chosen, and the price will be $1,008 per hour.

For Inspector, the Power BI Embedded sounds very valid option. Of course, the pricing might be too high, and the more accurate calculation must be done if the Power BI Embedded is chosen to be used. This mean calculating the potential peak renders per hour in potential customers. Of course, it should be noted that using ready visualization platform might mean much less development hours for own visualizations and possibility to focus on to the core knowledge of InSolution. Nevertheless, Power BI Embedded sounds so interesting that the capabilities of it with the node-based flow data is tested in the next chapter.

4. FLOW MONITORING SYSTEM

The necessity of new production flow application in current market situation is discussed in the chapter 3. Based on that, there is room for new, easy to use and adapt flow monitoring and traceability system. It is important for Inspector that it can be adapted to manufacturing systems without major modifications and breaks. This also means that flow monitoring system of Inspector could be used even if customer has their own automatic system running. The focus of Inspector flow monitoring system is in monitoring of production flow and traceability. Also, scalability of Inspector should be as easy as possible which means that also collected data must be as simple as possible.

In this chapter, the decisions about Inspector flow monitoring is made. The possible KPIs are developed and discussed, while the simple node-based data is formed. Also, deciding the used visualization methods are made by comparing the own implementation and Microsoft Power Bi Embedded. Finally, the future development ideas are provided.

4.1 Simple Data from Nodes

To get the material flow data from the manufacturing line, well-designed methods for the data collecting is needed. Using the core knowledge of InSolution, the obvious decision is to use PLCs for sending flow events to Inspector software. This means, that when reader attached to the PLC detects the flowing item, PLC sends the item information to Inspector, which then stores the data to the SQL database. The detecting method can vary from barcodes to the RFID based in the customer needs. For the flow analyses, only the structure of the information is meaningful. In this system, every reader and PLC combination is called a node.

The base idea for the monitoring system of Inspector was to make the adding of new nodes as easy as possible. This also means that the data send by the nodes must be as simple as possible. After designing the utilization of the monitoring system, the data form is decided to contain item or batch serial number, the node identifier for identifying the event source, and the timestamp of the event. The data event structure is presented in the table 17.

Table 17. *The event structure sent by the node after detecting an item*

Property	Definition
Item or batch (lot) serial number	Item or batch unique identifier used for tracking
Node identifier	Node unique identifier to inform the current location of the detected item
Timestamp	Timestamp of the item detection

Briefly, the item detection happens in the following way. First, the reader detects the item and reads the identifier. Then, the PLC forms the data by adding the node identification and timestamp. After that, the data is sent with an event to be stored to the database. To make the data handling faster, there is already initial version of the database developed for the Inspector which creates table called Transactions based on the data events. The Transaction table connects the consecutive events of the same item to the same database row to form single transaction between nodes. There is also Edge table to collect the cumulative data of the transactions between two nodes to make data handling more efficient and faster. Inspector can also save information about the items for allowing creating the item based KPIs and achieve better traceability. This is important in many discrete manufacturing and job shop plants where multiple different products are manufactured with the same machines.

By keeping the node-based data simple, adding the nodes is very easy. Now node can be configured to system only by adding equipment to the node location, adding unique identifier to the node and adding the node information to the Inspector database, and then node-based events are ready to be send.

4.1.1 Node types

Even though all the nodes in the system sends the data in similar format, the node types must differ for better analysis of the manufacturing flow and the system performance. There are 5 different kind of nodes designed which are a value-adding node, a buffer node, an output node, a checkpoint node and a scrap node. These types work similarly but has own functionality in analyses.

The value-adding node presents a cell, device or machine which is adding more value to the production item. Usually the value-adding node can contain only one item at a time. The buffer node is a node type where the value of the items is not increasing. The items are only waiting for proceeding. Typical buffer nodes are input and output buffers of machines or production cells, storages, plant floors or other places to store the items. With analysis of the buffer nodes, software can provide information about WIP or material items.

Output node is used to check out the items from the system. The item life time in system starts when it is first time detected at some node and ends when it is detected at the output node. There can be multiple output nodes and detecting the item in any of them makes the item ready. After the item is completed, couple of key indicators can be used to analyse the time of the item in the system. One of the most relevant indicators is a throughput time. Also, the path of the item can be shown from the start to the end, for example with directed graphs. If the item is detected as faulted, broken or scrapped, it can be informed to the system by moving the item out through scrap node. The scrap node works similarly than output node but software marks item as scrap after receiving the information from the node.

The checkpoint node is a dummy node only to inform that the item has been detected at the node. The item is usually moving, and it is not stopping at the node and the value of the item is not increased. For example, the node can be placed in the middle of the conveyor.

When a new node is added to the system, it must be presented in the Inspector. At the first point, this is done by InSolution developers but later this could be done from user interface. When node is added, only node name and node type are needed. If location of the node is known, also coordinates can be added for traceability and directed graphs.

4.2 Selected KPIs, ISO 22400:2 KPI Descriptions and KPI-ML Schemas

To offer added value for multiple customers, some basic KPIs must be formed using the help of ISO 22400:2. Also, KPI-ML schemas are created for possible integrations with customer own software like ERP or MES. The 8-step iterative closed loop model of Rakar et al. (2004) is used as a guide line for KPI design and creation. Of course, the KPIs are not designed for the own company so monitoring and reviewing of the KPIs is a challenge and the future acts for the KPIs are based on customer feedback. Nevertheless, because InSolution is using lot of focus and time on KPIs, lot of effort should be put on KPI analyse and development.

First, the designing of the possible KPIs starts by defining the goals and objectives for the monitoring system. The goal is to offer useful and important information about manufacturing flow in the discrete manufacturing plants. KPIs should be general enough to fit for multiple customers. The KPIs should be formed from the very simple event-based data which is adding own challenges to development. Still, by analysing the designed data format, multiple possible KPIs are identified. For example, utilization of the machine can be calculated when we know when the item entered and move off from the machine. Of course, then we must predict that readers are used before and after machining, meaning for example usage of input and output buffer nodes for the machine. Other possible KPIs are for example scrap ratio, throughput time, cycle time, WIP inventory, output rate, process time, queue time and move time.

Because the target is to focus on production items instead of machines, KPIs related to machines, like utilization, are dropped away. Current Inspector version can measure these KPIs already using different kind of data. Of course, it would be interesting to compare values formed from different data sets and see if there are differences.

The selected KPIs are average inventory, scrap percentage, throughput time, output rate and process, buffer and move time. With these KPIs, the customer should be able to plan the production and detect the possible bottlenecks. Of course, the KPI parameters are selectable to allow creating KPI average inventory for single buffer or KPI throughput time between specific nodes.

To present and communicate used KPIs to customers, ISO 22400:2 KPI descriptions are found to be effective and clear way. The KPI description gives lot of information about KPI and understanding of KPI is easier with it. Therefore, KPI descriptions are created for every decided KPI. However, effect model diagrams of KPI descriptions are not found effective enough and implementing them is not done at this point.

4.2.1 Average Inventory

Average inventory is used for analysing inventory peeks and lows in different time. By monitoring the inventory levels, better material flow can be designed. It is expensive to keep too much items in the inventory and finding the balanced inventory level is very important. Also, different analysis can be made by comparing inventory levels with the sales volume, for example. Should be noted that Inspector handles all the items in the system as WIP, because when the item is first time detected at some node, it is added to Inspector monitoring system and there is not difference between WIP and raw material. It could be possible to separate the material and WIP by analysing the route so that after first visit at value-adding node, the item is marked as WIP. However, this is not implemented at this point.

Usually in the literature, the average inventory value is given as monetary amount and the used formula is quite simple. This is provided in the equation 1.

$$Average\ Inventory = \frac{Beginning\ Inventory + Ending\ Inventory}{2} \tag{1}$$

In this system, the average inventory value is given as item count. The item count in the beginning of the time range and at the end of the time range might not describe the real average inventory value truthfully and therefore other way of calculating the average inventory is created. For example, the figure 47 presents simple example of problematic with the equation 1.

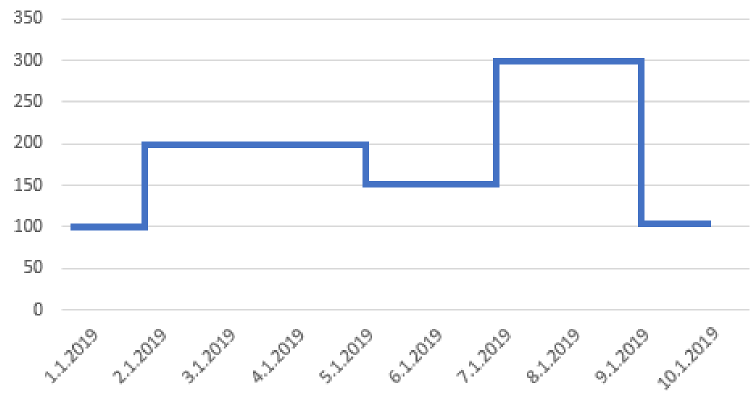


Figure 47. *Simple example of inventory levels for 1.1.2019 – 10.1.2019*

In the figure, transaction event to the buffer happens when the inventory level goes up and transaction event from the buffer happens when the inventory level goes down. As can be seen, the average inventory using equation 1 is 100 pieces but the real average inventory in the time range is much higher. When the weighted arithmetic mean is calculated, more detailed value can be got. The values are the buffer levels at the transaction point and the weights are time in seconds which the buffer level stays constant. In this simplified example, the transactions happen only once a day at the same time. When using the values in the figure 47, we get the weighted arithmetic mean presented in the equation 2.

$$\begin{gathered} Average\ Inventory = \frac{100 * 86400 + 200 * 259200 + 150 * 172800 + 300 * 172800 + 100 * 86400}{86400 + 259200 + 172800 + 172800 + 86400} \\ Average\ Inventory = \frac{146880000}{777600} \\ Average\ Inventory = 188{,}8889 \ldots \approx 189\ (pcs) \end{gathered} \tag{2}$$

So, the more realistic value for the average inventory is 189 pieces. The difference to value calculated with equation 1 is huge and therefore, the weighted arithmetic mean is decided to be used for average inventory. The formula is provided in the equation 3

$$Average\ Inventory = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n}{w_1 + w_2 + \cdots + w_n} \quad (3)$$

where n is amount of buffer level changes during the time range, x is a buffer value and w is a weight which is the time how long the buffer level stays constant in seconds.

The idea is to get the average inventory in the specific time for wanted items. Wanted buffers can also be selected if there is need to know the buffer levels only at the certain buffers. Of course, getting the total average inventory for every item type in all the buffers is also possible. The table 18 presents the KPI description formed in ISO 22400:2 KPI description style.

Table 18. *ISO 22400:2 style KPI description for average inventory*

KPI Description	
Content	
Name	Average inventory
Description	The average inventory in a given time frame.
Scope	All item types, specific item type
Formula	$Average\ Inventory = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
Unit of measure	pcs
Range	Min: 0, Max: Unlimited
Trend	The smaller the better (Inspector is handling WIP parts)
Context	
Timing	On-demand, periodically
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Beginning and ending time of calculation can be selected. Specific WIP buffer can be selected. A specific item type can be selected.

There is not direct equivalent for average inventory KPI presented in the ISO 22400:2 but there is mention of average inventory in KPI named inventory turns. For Inspector, at the beginning it is more reasonable to offer KPI for average inventory than for inventory turn.

4.2.2 Scrap Ratio

The scrap ratio informs how big part of the produced items become scrapped. Scrapped items are usually waste or at least they cause more expenses via rework. It is important to know how much there is scrap items compared to the total production amount. The scrap ratio is designed to be calculated in the Inspector using the equation 4.

$$Scrap\ Ratio = \frac{Scrap\ Quantity}{\text{Total Produced Quantity}} * 100\% \tag{4}$$

The scrap ratio can be calculated for specific time frame for selected item types. Also, the total scrapped amount for all the items can be calculated. Because the monitoring system have special scrap node which is used to inform about part scrapping, also scrap amount after some specific nodes can be requested. This allows detecting faulty operating machines for example. The table 19 presents the KPI description for scrap ratio.

Table 19. ISO 22400:2 style KPI description for scrap ratio

KPI Description	
Content	
Name	Scrap Ratio
Description	The scrap quantity (SQ) compared to the total produced quantity (TPQ).
Scope	All item types, specific item type
Formula	$\text{Scrap Ratio} = \frac{SQ}{TPQ} * 100\%$
Unit of measure	%
Range	Min: 0 %, Max: 100 %
Trend	The smaller the better
Context	
Timing	On-demand, periodically, real time
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Beginning and ending time of calculation can be selected. Specific item type can be selected. Scrap after some specific node can be selected.

ISO 22400:2 also defines KPI description for scrap ratio. The same equation for the KPI is used in ISO 22400:2 than in designed flow monitoring system.

4.2.3 Throughput Time

Throughput time tells the needed time for the item to pass through the manufacturing process. This KPI gives information about process performance. Usually, this is calculated by combination of item process time, move time, inspection time and queue time. In Inspector, the item is checked in to the process in the first node where the item is detected. When the item is completed, it is checked out from output node. Therefore, the throughput time is calculated to be the time difference between the first detection of the item and the detection of the item at the output node. (Equation 5).

$$Throughput\ Time = \text{Output Node Time} - \text{First Node Time} \tag{5}$$

Throughput time can be calculated as average for all the item types, or as average for single item type, or for a specific item using the item serial number. The KPI description is provided in the table 20. ISO 22400:2 consist KPI called throughput rate but it is calculated order-based instead of item-based.

Table 20. *ISO 22400:2 style KPI description for throughput time*

KPI Description	
Content	
Name	Throughput time
Description	Time required for item to pass through a manufacturing process. Throughput time is difference between output node time (ONT) and first node time (FNT)
Scope	All items (average), specific item type, specific item
Formula	Throughput time = ONT - FNT
Unit of measure	time (s, min, h, d)
Range	Min: 0, Max: Unlimited
Trend	The smaller the better
Context	
Timing	On-demand, periodically
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Some specific item can be selected. Average time is provided for multiple items and exact time for a specific item

4.2.4 Process, Queue and Move Time

The next three KPIs describe different times that an item spends in different areas of production. These KPIs are process time, move time and queue time, and these are not presented in the ISO 22400:2. However, these KPIs can offer important information, at least when they are analysed next to each other. For example, comparing process time and queue time can offer information about how big part of total manufacturing time the item is only waiting for processing. To be able to get these KPIs with Inspector, every machine needs both input and output readers to detect exact times when the item arrives and when leaves the machine.

Process time KPI describes how much time an item spends in machines, devices and other processing nodes. In Inspector, these nodes are value-adding nodes. In turn, queue time is thought as time spend in buffer nodes. The move time describes how much time is needed for transportations between different nodes. KPI descriptions for process, queue and move time are presented in the tables 21, 22 and 23. The closest KPI in ISO 22400:2 is production process ratio which combines actual process times in work stations and compares them to the total throughput time of the item. Throughput time KPI is combination of process time, queue time and move time.

Table 21. *ISO 22400:2 style KPI description for process time*

KPI Description	
Content	
Name	Process time
Description	Time required for raw material conversion to the ready item. Total time spent in value-adding nodes (N_nT_n).
Scope	All items (average), specific item type (average), specific item
Formula	$Process\ Time = \sum_{i=1}^{n} N_iT_i$
Unit of measure	time (s, min, h, d)
Range	Min: 0, Max: Unlimited
Trend	The smaller the better
Context	
Timing	On-demand, periodically
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Some specific item can be selected. Average time multiple items

With the data got from the nodes, multiple variations for process time KPI can be developed. The process time can be got for single item or single item type, or for multiple item types. For multiple items, the value of process time KPIs is average of process times of different items. The used equation is provided in equation 6

$$Process\ Time = \sum_{i=1}^{n} N_iT_i = N_1T_1 + N_2T_2 + \cdots + N_nT_n \tag{6}$$

where n is number of value-adding nodes and N_nT_n is time spend in a value-adding node.

Table 22. *ISO 22400:2 style KPI description for move time*

KPI Description	
Content	
Name	Move time
Description	Time required for moving items between nodes. (M_nT_n) The total move time in the process for single item, time between two nodes for single item or average time between two nodes can be calculated.
Scope	All items (average), specific item type (average), specific item
Formula	$Move\ Time = \sum_{i=1}^{n} M_iT_i$
Unit of measure	time (s, min, h, d)
Range	Min: 0, Max: Unlimited
Trend	The smaller the better
Context	
Timing	On-demand, periodically
Audience	Operator. supervisor, management
Production methodology	Discrete, batch
Notes	Total moving time for a specific item or average moving time for an item type. Average move time for all items. Average time between two nodes for single or all items.

For the move time, many variations can be created. Move time through the whole process or move time between two nodes for specific item or item type can be created. Also, average move time for all the items can be formed. Average move time between two nodes can also be queried. The equation 7 shows how the move time is formed.

$$Move\ Time = \sum_{i=1}^{n} M_iT_i = M_1T_1 + M_2T_2 + \cdots + M_nT_n \tag{7}$$

There, n is number of item transactions and M_nT_n is move time between two nodes.

Table 23. *ISO 22400:2 style KPI description for queue time*

KPI Description	
Content	
Name	Queue time
Description	Time spent in buffers. (B_nT_n) For single item, for specific item type or for all items.
Scope	All items (average), specific item type (average) or specific item type
Formula	$Queue\ Time = \sum_{i=1}^{n} B_iT_i$
Unit of measure	time (s, min, h, d)
Range	Min: 0, Max: Unlimited
Trend	The smaller the better
Context	
Timing	On-demand, periodically
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Total queuing time for specific item or average queuing time for item type. Average queuing time for all items.

Queue time can be formed many ways. Queue time can be got for a single item during the whole manufacturing process or in a specific buffer node. Then, the average queue time for a single item type or for all the items can be formed. The equation 8 shows the used formula

$$Queue\ Time = \sum_{i=1}^{n} B_iT_i = B_1T_1 + B_2T_2 + \cdots + B_nT_n \tag{8}$$

where n is number of buffer nodes and B_nT_n is time spend in a buffer node.

4.2.5 Output Rate

Output rate KPI tells the total amount of ready items in a time frame. In Inspector this means the number of items passing through the output nodes in a configurable interval. The value can be calculated for single item type or for multiple item types. The description for the KPI is presented in the table 24.

Table 24. *ISO 22400:2 style KPI description for output rate*

KPI Description	
Content	
Name	Output rate
Description	Number of ready items in a time frame.
Scope	All item types, specific item type
Formula	
Unit of measure	pcs
Range	Min: 0, Max: Unlimited
Trend	The larger the better
Context	
Timing	On-demand, periodically
Audience	Operator, supervisor, management
Production methodology	Discrete, batch
Notes	Can be calculated for specific item type or for multiple item types. Time frame can be changed.

By analyzing and monitoring the output rate, changes in the production can be detected. The output rate can vary for multiple reasons. Examples are machine breaks and maintenance, changes in the employees or changes in the customer and production orders.

4.3 Traceability with Inspector

Inspector will allow tracking the specific item movement in the manufacturing area by querying the data with the item unique identifier. This is possible because the Inspector saves the movement data from the nodes. Then, it is possible to know in which node the item is currently or to get the historic data of the item movements. The plan is to visualize the historic traceability data with the directed graphs. The directed graph has nodes and edges with directions. The edges are vectors pointing from node to node to illustrate the transaction of the item between two nodes. The figure 48 presents possible directed graph of 2 items in the manufacturing area.

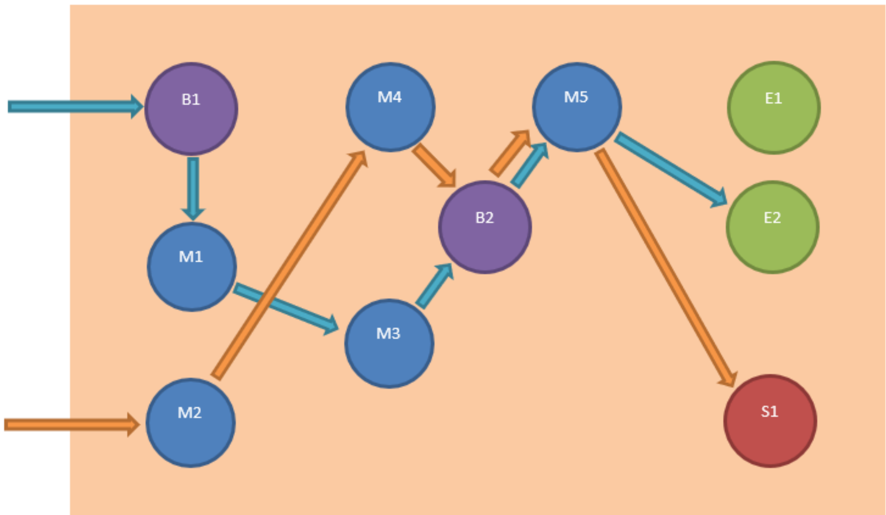


Figure 48. *Plan of directed graph of two different item workflows*

The example manufacturing area consists 2 buffer nodes (B1 and B2), 5 value-adding nodes (M1-M5), 2 output nodes (E1 and E2) and a scrap point node (S1). Movements of the first item are visualized with blue vectors and the movements of the second item with orange vectors. With directed graph, it is possible to see the workflow for items. In the figure 48, the blue item is first detected at the buffer 1 and the life time of item starts there. Finally, after multiple node visits, the blue item is booked out from the system in the output node 2. The orange item is first detected at the value-adding node 2 and finally booked out as scrap after value-adding node 5.

The idea is to provide also other movement data with the directed graphs. For example, the different routes for the specific item type can be visualized to identify and analyze different patterns of production flow. This is useful for example when there are multiple parallel workplace nodes because then it is possible to see which workplaces are used the most. The amount of transactions between nodes can be visualized so that the higher amount is visualized with a thicker vector. The figure 49 presents the planned directed graph for single item type.

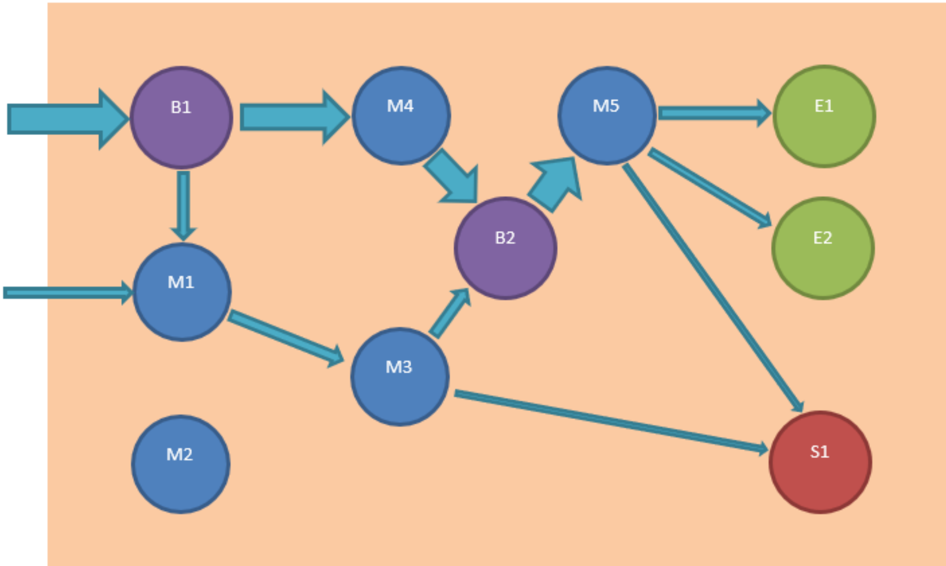


Figure 49. *Directed graph for item type is used to visualize the alternative routes and item amounts in each route*

As can be seen from the thicker vector, the items are mainly introduced to the system via buffer 1 but sometimes items are also going directly to the value-adding node 1. Then, most of the items are going through the value-adding node 4 but also route via value-adding nodes 1 and 3 is possible. This means that in the node M4 some operations can be done to the items which are possible in M1 and M3 combined. This might mean for example grinding of the tube-shaped item where outside can be grinded with M1 and inside with M3, but M4 can do the both steps. Nevertheless, using the route M1-M3 is causing more scrap items than route through M4. Finally, after M3 or M4, the parts are going to buffer 2 and then to the value-adding node 5. After that the parts are booked out of system with output nodes E1 and E2 or marked as scrap with the scrap node S1.

4.4 Interface to the Backend

To analyse and visualise collected node-based data, the data must be stored and queried from the database. Implementing flow monitoring to Inspector happens in two parts and this thesis is the higher-level of the implementation which means forming, analysing and visualizing the KPIs and the traceability data. The lower-level means data collecting and storing in the database and handling database queries. To allow data transfer between these levels, common interface is developed based on the developed and chosen KPIs of this thesis. The figure 50 presents the designed architecture of Inspector flow monitoring system and the coloured areas are linked with this thesis.

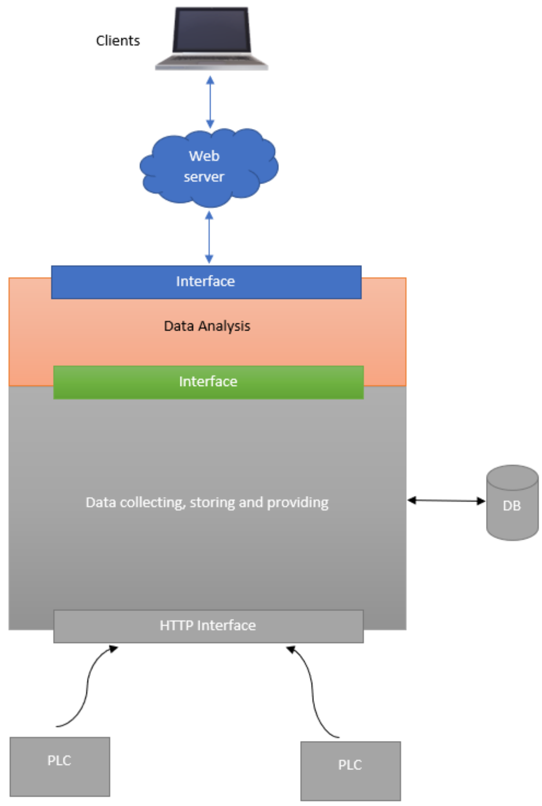


Figure 50. *Designed architecture of Inspector flow monitoring system. The interface between data analyses and data collecting is marked with the green color.*

The interface to provide data between data analysis module and data collecting, storing and providing module is marked with green colour. At the designing phase, the interface was thought to has several functions to get the needed data for KPIs. The idea is to get the data so that it is usable for KPI creation, traceability analysis and visualization. This means that the data is not coming as ready, but it needs processing in the data analysis layer of Inspector. From the programming perspective, the interface is spitted to two interfaces based on the functions.

First programming interface is based on KPI values. At this point, knowing the interface and designed nodes better, move time is not implemented but stays in the future. These functions are presented in the table 25 with short descriptions. Better function presentations can be found from Appendix B where parameters and return values of functions are also presented. The preliminary name of the interface is IFlowAnalyticsCore, and the name is following Inspector coding guide.

Table 25. *IFlowAnalyticsCore-interface is used for getting node-based data from the database to form KPI values and visualize them*

Function	Description
GetAverageInventory	Function returns average inventory size in a given time frame for wanted item types. Function returns average inventory separately for every item type. Resolution of return value can be selected. (min, hour, day, week, month)
GetScrapPercentage	Function returns scrap percentage in a given time frame. Scrap after specific node or scrap of specific item type can be selected.
GetThroughputTimeForSingleUnit	Function returns throughput time for a single item based on an item serial number. Start and end node can also be selected to get the partial throughput time.
GetThroughputTimeForMultipleItems	Function returns average throughput time for selected item types. Time range can be selected. Start and end node can also be selected to get the partial throughput time.
GetProcessTimeSingleUnit	Function returns process time for single item based on an item serial number.
GetProcessTimeForMultipleItems	Function returns process time for selected item types. Time range can be selected.
GetBufferTimeSingleItem	Function returns queue time for a single item based on serial number.
GetBufferTimeMultipleItems	Function returns queue time for selected item types. Time range can be selected.
GetOutputRate	Function returns output rate in the given time range for a single item type or for all item types. Resolution of return value can be selected. (min, hour, day, week, month)

For an example, the better presentation of function is provided in the table 26. Other functions can be found from the Appendix B.

Table 26. Presentation of function GetAverageInventory

GetAverageInventory			
Used to get average inventory size at given time period. Using different parameters, including only specific buffers or item type is possible			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
type	string	yes (“”)	Wanted item type. Default value for all item types.
buffers	List<Guid>	yes (null)	Wanted buffers. Default value for all buffers.
resolution	TimeResolution	yes (0)	Wanted time resolution. Default value for all.
Return value	**Return value data type**	**Return value default**	
	Dictionary<Guid, List<float>>	new Dictionary<Guid, List<float>>()	Average inventory size for item types (dictionary) in different times based on resolution (list)

The second programming interface is for querying node-based traceability data from the database to analyze and form visualizations of production flow, for example with directed graphs. The interface is called IFlowRoutesCore. The functions are presented in the table 27. More detailed function descriptions are provided in the Appendix C.

Table 27. IFlowRoutesCore-interface is used for getting node-based data from the database to form traceability data and visualize production flow

Function	Description
GetRouteForItem	Function returns flow route from node to node for single item based on item unique identifier.
GetRoutesForItemType	Function returns flow routes from node to node for the items of specific type.
GetRoutes	Function returns flow routes from node to node for the all the items.
GetNodeItems	Function returns items in node at the wanted time.

4.4.1 Another thesis: Device Solution for Flow Monitoring System

As mentioned earlier, the flow monitoring system of the thesis is split to two parts. Olli-Petteri Hirvonen writes about the backend and database operations with subject "Designing a Backend Service and Database Structure for Monitoring Discrete Manufacturing Systems". Nevertheless, there wasn't any kind of doing together instead of IFlowAnalyticsCore-interface specification. The interface is designed based on the chosen KPIs.

Thesis of Hirvonen is focusing on different data modelling methods for data storing. The thesis compares the models and performance of Petri nets and directed graphs. As a conclusion, the directed graphs model is selected to the database storing method.

4.5 Possible visualization libraries for Inspector

If it is chosen to create visualization with own implementation, using the ready libraries for rendering the charts should be used. After analyses, two options for visualization libraries are found. For creating the charts, Chart.js is valid option. For directed graphs to visualize traceability data, Cytoscape.js is found to be suitable library.

Chart.js is JavaScript library for creating flexible and simple charts. It contains 8 different chart types and they are rendered with HTML5 allowing interactivity with all modern browsers. The charts are designed to be responsive to allow designing the visualizations for every screen size. Also, animation of charts is possible. The Chart.js is working with MIT license making it possible to use also in the commercial applications. (Chart.js 2018)

Cytoscape.js is JavaScript library for network theory. It allows creating interactive network graphs and it supports all the modern web browsers. Cytoscape.js is interactive by default and animations are also possible. Cytoscape.js uses MIT license and therefore it is usable also in the commercial applications. (Cytoscape.js 2018)

4.6 Choosing the Visualization Platform for Flow Monitoring System

At the start of the thesis, the idea was to create own solution for visualizing discrete production flow data if ready solutions are not found. After exploring the other solutions, Microsoft Power BI Embedded seems to fit the Inspector well. Before more detailed cost calculations and implementations, some tests are performed to see if Power BI can really save development time by creating the data visualizations instead of own solution.

The possible architecture of Inspector flow monitoring system with the Power BI Embedded is presented in the figure 51. The new Power BI changes are presented with the orange colour. The function of Inspector with Power BI is quite simple. First the clients are requesting an embed token from the Inspector server. Then, the Inspector server requests the token from the Power BI server and sends it to the client. After that, the client can access the visualizations created with the Power BI without Power BI user account. The visualizations are created with the Power BI tools and then published to the Power Bi server. Power BI server queries the data from the database using interface located between Data Analysis and Data collecting, storing and providing layers.

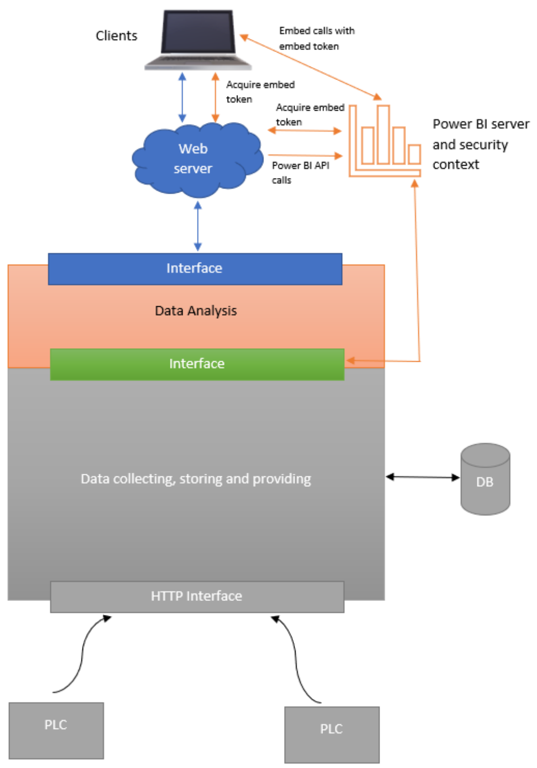


Figure 51. *Architecture of Inspector flow monitoring system with Power BI Embedded*

Before implementing Power BI Embedded architecture, the benefits of ready visualization and Power BI platform must be tested and compared to the own implementation. Therefore, both Microsoft Power BI and own implementation are tested to create example visualizations from realistic looking data. To get the needed data, a simulator to implement the designed IFlowAnalyticsCore-interface and IFlowRoutesCore-interface is constructed. This is done with C#-language to benefit the real implementation of Inspector. The simulator is simulating the Data Analysis and Data collecting, storing and providing modules of the designed architecture.

First, the Data Analysis layer queries the data from the imaginary database through real interfaces. The queried data is then formed using input values. The data is based on deliberated starting values and then little randomization is added to create different, realistic, data sets. After data is returned to the Data Analysis layer, the data is formed to CSV-file which can be read by Microsoft Power BI and by an own server implementation. The visualization server is done with Node.js and the used visualization library is Chart.js. The idea is to compare Microsoft Power BI and Chart.js implementation based on visualization creation ease and effectiveness. For this, the charts are kept as default as possible. Of course, by customizing the visualizations can be formed differently in both solutions. But to get the idea and examples for comparing, the charts are kept as basic as possible. The example charts are done for average inventory data. The production type is thought to be a job shop and there are multiple item types manufactured in the same plant.

First, a line charts for item types in inventory are formed with Chart.js and with Microsoft Power BI. The start time is 20th of October and end time 13th of December. The used resolution is a day which means that the average inventory value is returned day by day. The charts are presented in the figure 52 so that Chart.js chart is upper and Microsoft Power BI chart is lower.

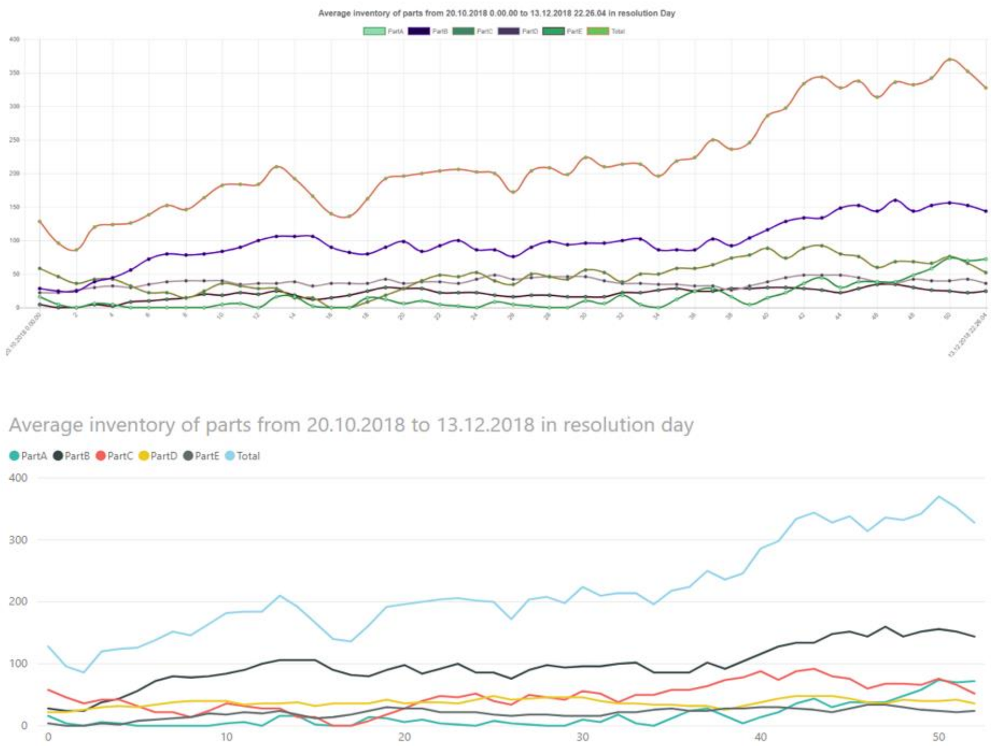


Figure 52. *Average inventory for item types with resolution day visualized with Chart.js (upper) and Microsoft Power BI (lower)*

As can be seen, both charts are clear by default. There are little differences with the labels and titles, but these can be configured later. Both charts are also interactive by default. The figure 53 presents the interactivity in the charts when only PartA and Total are selected from both charts. The selection happens by clicking the label of the item type, in this case PartA and Total.

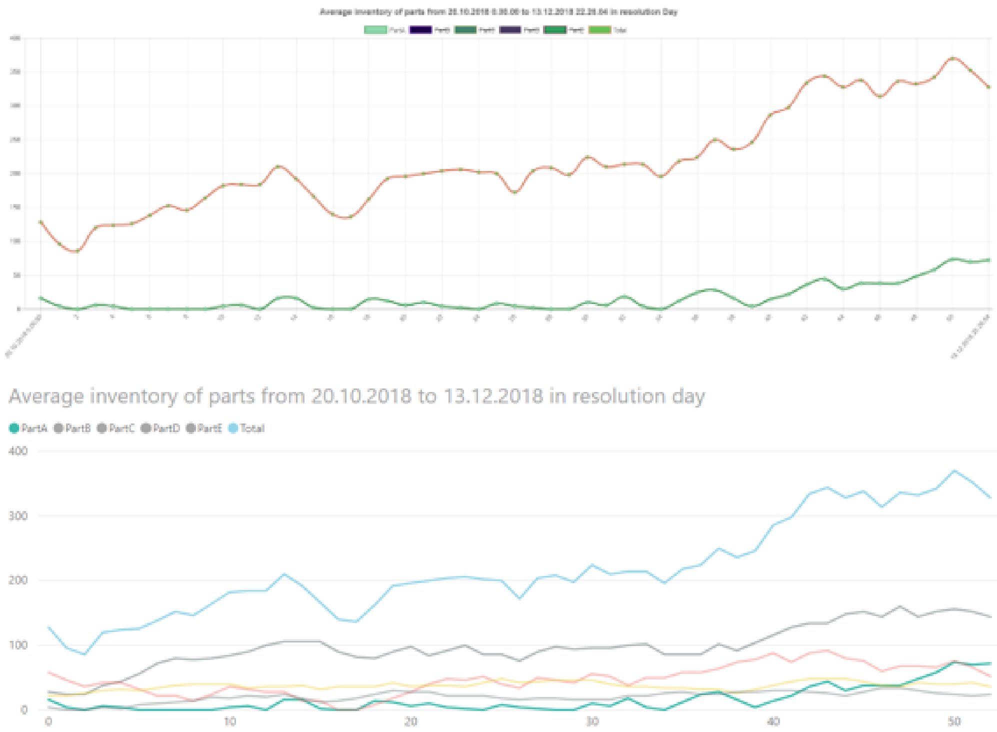


Figure 53. *Average inventory filtered with Chart.js (upper) and Microsoft Power BI (lower)*

The Chart.js chart is filtered so that only selected values are shown while Power BI only highlights the selected values. The Chart.js way is better for Inspector point of view because the selected values are more visible, and this makes the analyse easier. The figure 54 presents the same kind of average inventory line charts for specific item type in a specific buffer over the year 2018 with resolution month.

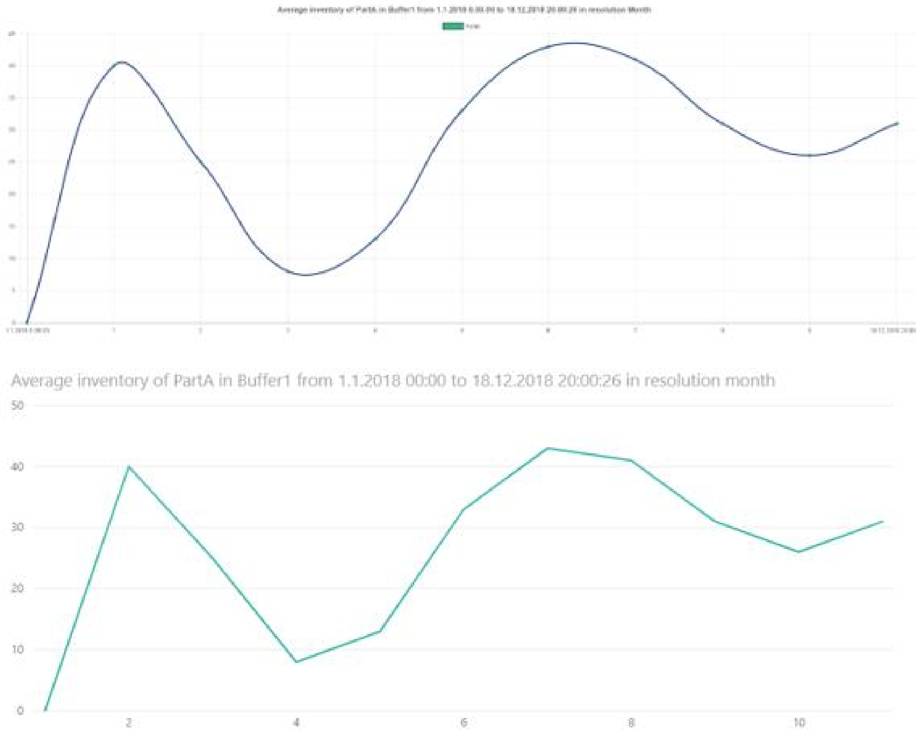


Figure 54. *Average inventory of PartA in the Buffer1 during the year 2018 with resolution month using Chart.js (upper) and Microsoft Power BI (lower)*

As can be seen, the Power BI uses sharper edges by default. Other vice, the lines are quite similar and there are not much visual benefits using the Power BI. Figure 55 presents the pie charts for the average inventory levels of different item types at the time range 1.1.2018 00:00:00 – 18.12.2018 20:00:26.

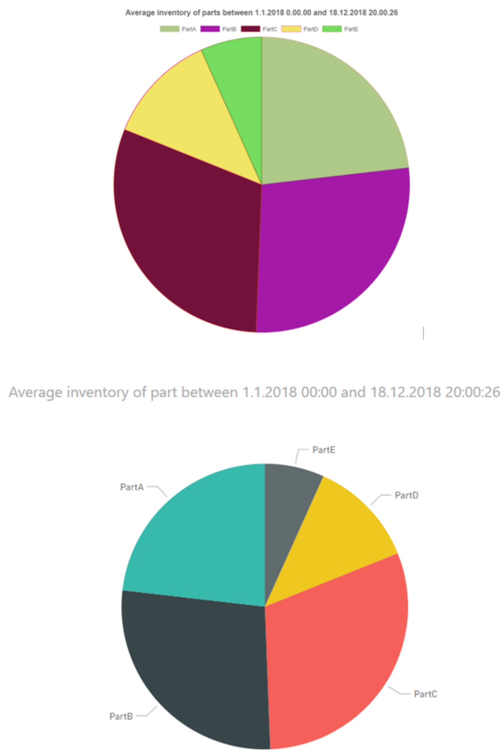


Figure 55. *Pie chart for average inventory of different item types using Chart.js (upper) and Microsoft Power BI (lower)*

Also, the pie charts are quite similar, but the labelling differs by default. The differences are still small, and the visualization aspects are not big enough to justify using the paid visualization platform.

Creating the charts with the Chart.js is surprisingly easy and good manuals and help can be found easily. Also, using the coding language to read the data from the CSV allows flexibility much more than using the Power BI Desktop. In the Power BI, the data must be formed better, for example to the table, to allow creating the charts. For example, the CSV data used in the average inventory line charts is not suitable for Power BI as itself but with Chart.js rendering of the line chart with the CSV data is possible. The CSV format is presented in the program 1.

```
PartA;20.10.2018 0.00.00;13.12.2018 22.26.04;Day;16;4;0;6;4…
PartB;20.10.2018 0.00.00;13.12.2018 22.26.04;Day;28;24;24;38;44…
PartC;20.10.2018 0.00.00;13.12.2018 22.26.04;Day;58;46;36;42;42…
PartD;20.10.2018 0.00.00;13.12.2018 22.26.04;Day;22;22;26;30;32…
PartE;20.10.2018 0.00.00;13.12.2018 22.26.04;Day;4;0;0;4;2…
```

Program 1. *Example of used CSV-file between simulated data generator and visualization software*

The CSV file contains the values item type, start time, end time and resolution. Then the actual inventory values are listed. The Power BI could not draw the line chart because it is not possible to combine the value to the x-axis point. With Chart.js, it is possible to present the x-axis points easily. For Power BI presentation, the CSV file must be edited before use. Editing the CSV file with Power BI is surprisingly time consuming and therefore using the Excel is faster. This means that 2 different applications are needed to create the line chart with the Power BI. Creating the pie chart is easy also with the Power BI. Of course, should be noted that the CSV files are used only for simulation purposes, but these gives very good understanding about possible challenges with the Power BI data reading.

One benefit of the Power BI is that customizing the visualizations is easier when the customization, for example line widths and colours, can be done with the Power BI Desktop application. With Chart.js, knowing the CSS is necessary. Still, creating the CSS for Chart.js is quite easy, and lots of help can be found because Chart.js is widely used visualization library.

On the other hand, using the Power BI Desktop is not so easy than it should be considered as valid option for Inspector. It wants to sort fields wrongly and changing the sorting is not clear enough. Also, the application is quite time and resource consuming. Of course, by deeper training, the Power BI Desktop might become easier to use.

To summarize, the flexibility and interactivity of Chart.js is very important for Inspector flow data visualization and the usage of it is quite straightforward, which makes adopting expensive Power BI embedded unnecessary. The visualization of the data also looks lot like visualizations of the Power BI. Of course, by creating the visualizations itself, lot of time must be used for designing, customizing and learning. Anyhow, also Power BI Embedded would increase developing hours because using the Power BI Desktop is not easy enough and lot of training could be needed.

4.6.1 Selected technologies

Because the Power BI was not valid option and there were not any other suitable ready solutions, the flow monitoring of Inspector will be implemented itself. For the KPI visualization, Chart.js is found to be effective, easy and interactive library. For visualizing traceability data, Cytoscape.js is selected. These visualization libraries are fitting the Inspector current UI design because HTML5, Angular and JavaScript are already used. Because original Inspector uses C# as backend technology, this is also used with flow monitoring application. This allows integration and using of ready helper functions of current Inspector.

4.7 KPI Analyses, Traceability and Flow Monitoring with Inspector

By monitoring the discrete production flow, customers of InSolution can get multiple benefits. They can identify bottlenecks, remove waste by identifying reasons for scrap, design and implement suitable warehouse and inventory by identifying inventory amounts, detect peaks in throughput time of specific items and find the routes of items with traceability data. This chapter is giving examples of possible KPI and traceability visualizations with Inspector. Also, some benefits of the visualizations are discussed. Should be noted that the purpose of flow monitoring system is to provide added-value to the customers, not to collect data without any real targets.

By finding the throughput time, customers can identify average times for the item type to go through the production process which allows calculating possible production amounts. Users can also get histogram to detect peaks and bottoms to find the deviations in the throughput time. This can help to improve production and to get steadier production flow. For example, the figure 56 presents the throughput time of last two hundred PartA-type of items.

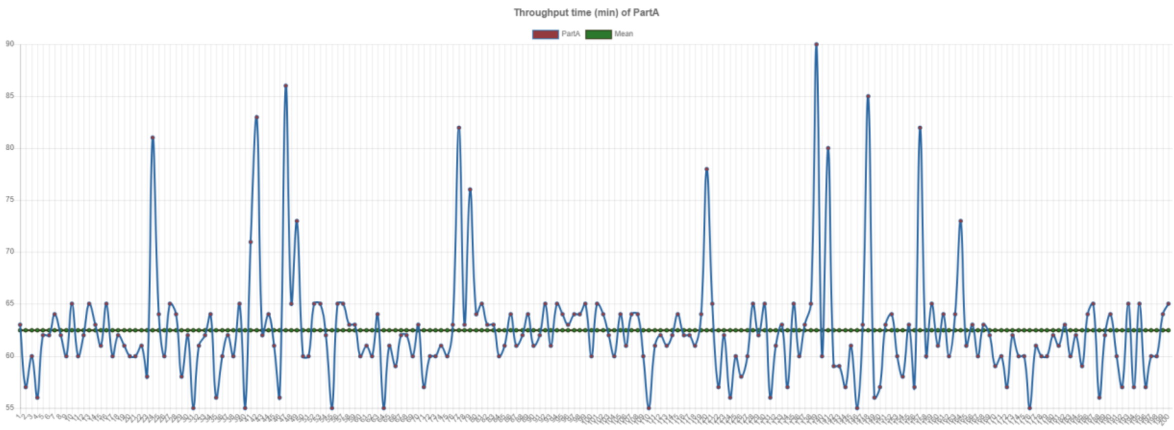


Figure 56. *Throughput time for last 200 PartA-type of items*

As can be seen, the mean throughput time is 62.5 minutes, but the variance is quite high. The fastest throughput time is 55 minutes and the lowest 90 minutes. By analysing the reasons for peaks and bottoms, the production flow can be developed to be more efficient. By rendering the histogram for throughput time, the frequency of different throughput times can be seen. Histogram is presented in the figure 57.

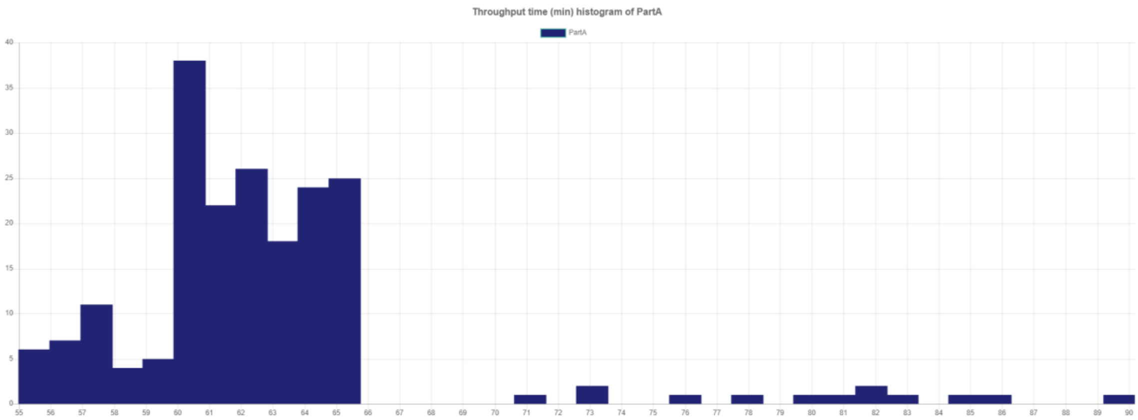


Figure 57. *Histogram of throughput times for last 200 PartA-type of items*

From the histogram can be seen, that the mode of the throughput time is 60 minutes. Most of the items of type A are going through the production between 60 and 65 minutes. This means that the designed and optimal through put time might lie between these values. For some reason, there are peaks for some parts causing deviations in histogram. This must be analysed in the production but for example maintenance tasks can cause delays for some items. Getting the data about the deviations is one of the big benefits of monitoring the production flow. To improve the process, variance between the items of same type must be developed to be as small as possible.

To find the reasons for scrapped items is important to reduce the waste in the production. With the flow monitoring, the scrap percentage can be got. Also, the scrap percentage after specific node can be identified. For example, if there are 3 parallel value-adding nodes, can scrap after these nodes be calculated to get the idea if some of the node is performing worse than the others. Example of this is shown in the figure 58 which shows the scrapping percentage of PartA-type of items after 3 parallel value-adding nodes. Parallel node means that in the production environment, any of these nodes can be used to process the same item operation.

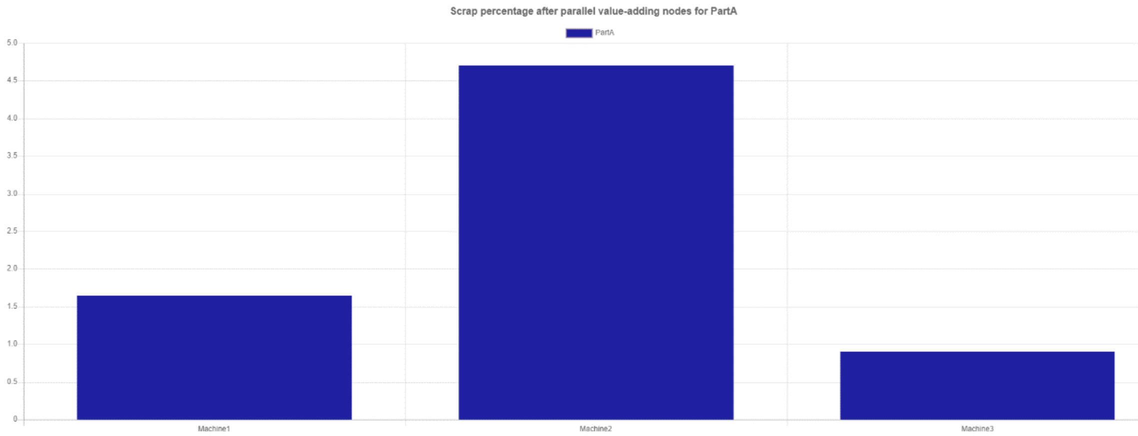


Figure 58. *Scrap percentage of PartA-type of items after 3 parallel value-adding nodes*

The scrap percentage after machine 2 is much higher than after machine 1 and 3. This might mean that machine needs maintenance, operators needs more training or quality control is higher at the end of machine 2. Finding information like could help to improve production performance.

4.8 Traceability and Flow Monitoring Visualization

Traceability and flow monitoring are visualized with Cytoscape.js by forming directed graphs. The directed graph edges, arrows or vectors, are formed between 2 nodes based on consecutive observations of the item. IFlowRoutesCore-interface is used to query node data from the core. By querying the data for single item with serial number, the route of the item can be visualized. Also, the timestamp of the node visit is displayed by selecting the node. Figure 59 shows the possibility of traceability visualization for the single item. The factory floor is having 3 value-adding nodes which are named machines, 2 buffers, 2 output nodes which are named loading docks and a scrap node. The data is created with the simulator so that it corresponds to the real factory environment. The simulator is randomly picking nodes in the route with configured rules which are simulating one possibility of the real factory environment.

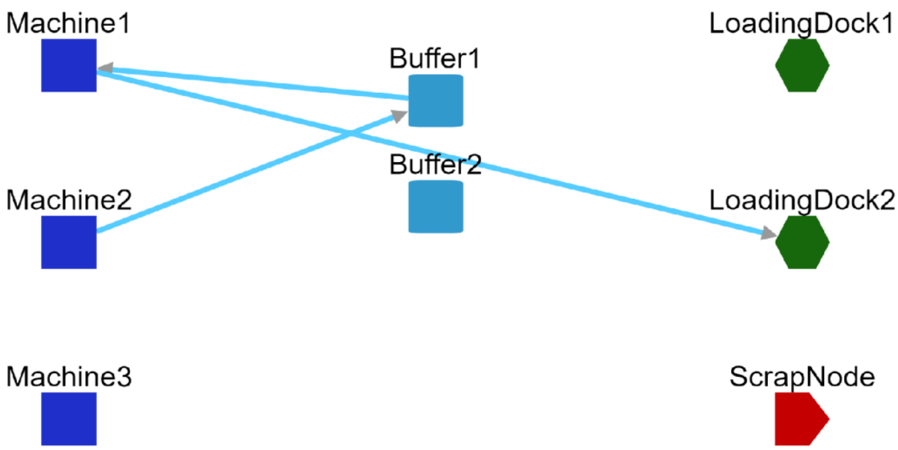


Figure 59. *Directed graph for visualizing single item flow through production process*

The figure 59 informs that specific item is first detected at the node machine 2. At this point, the item is registered to the Inspector and monitoring the flow of the item starts. After machining, the item moves to buffer 1. From the buffer 1 it moves back to value-adding node, this time to the machine 1. Finally, the item is booked out from the node loading dock 2. By selecting the node, information about time when the item is detected at the node is offered. This is presented in the figure 60. In the future, also different information can be got by selecting the node, for example items which are currently at the node.

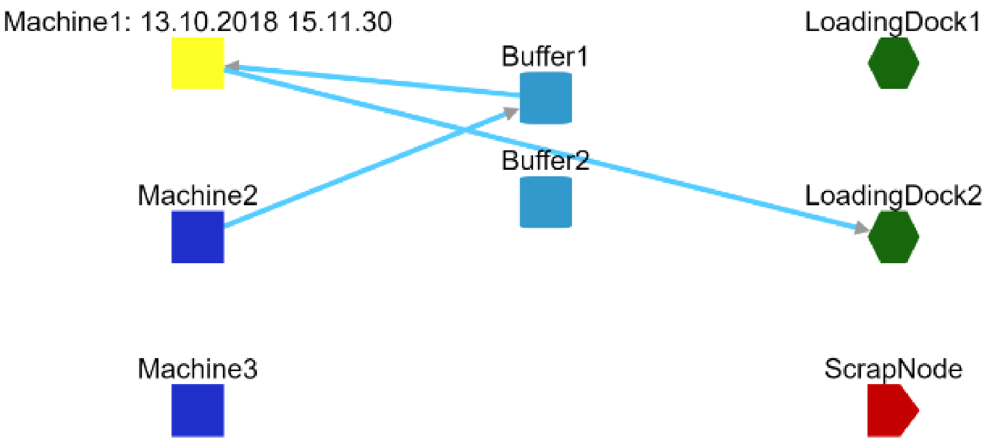


Figure 60. *By selecting the node, information about item detection time in the node is presented*

The item was detected at the node machine 1 at 13.10.2018 15:11:30. If the machines have also readers for machine input and output buffers, more data could be created. Moreover, using the directed graphs, information about flow of multiple items can be visualized. For example, the flow of specific type of items in wanted time frame can be queried. The figure 61 presents the flow of last 50 items of type PartA so that each item is coloured differently. The routes of the 50 items are simulated using randomization with specific rules.

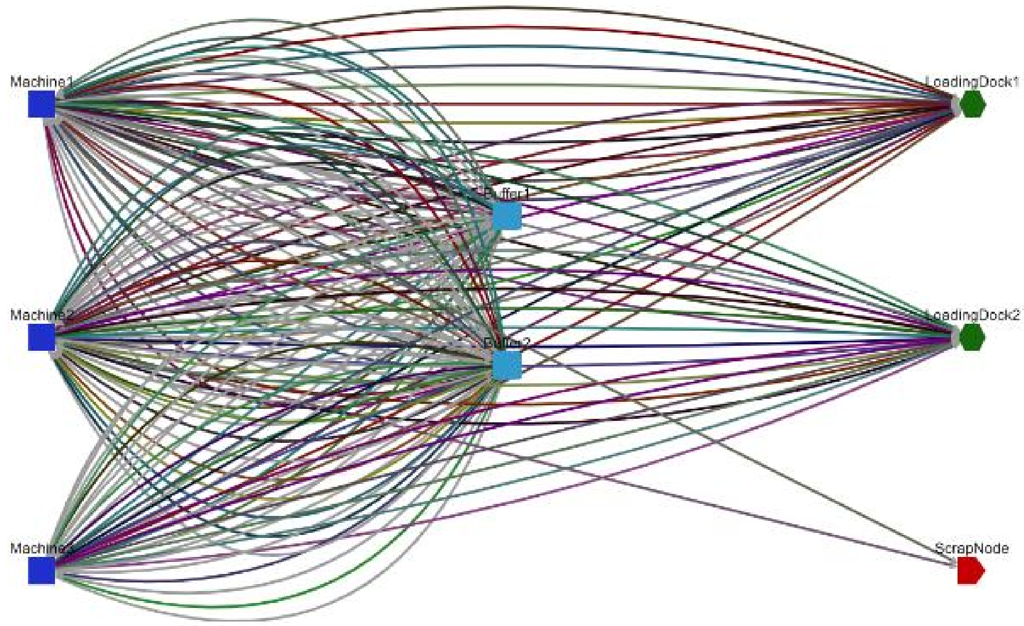


Figure 61. *Flow of last 50 items of PartA*

As can be seen, the directed graph will quite easily become almost unreadable. Therefore, figure 62 presents the same information so that the edges coming from the same node and ending to the same node are combined so that the vector thickness is presenting the quantity of edges. Thicker vector presents more item transactions between nodes.

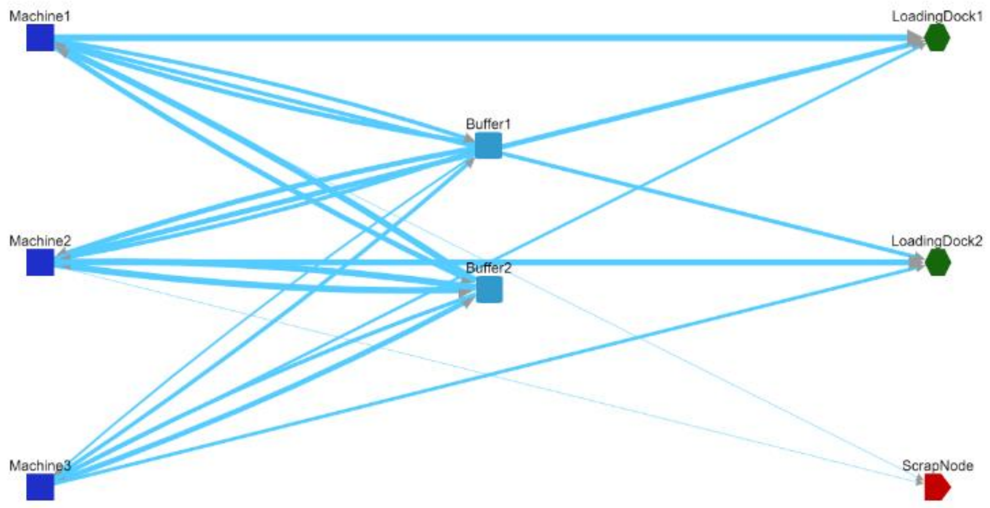


Figure 62. *Flow of last 50 items of PartA so that quantity of same edges are presented by the thickness of the vector*

Now the visualization is clearer. From the figure 62 can be detected, for example, that after node machine 1 there is more flow going to loading dock 1 than to loading dock 2.

Also, there are only a few items scrapped after machine 1 and machine 2. After machine 3, there are not items scrapped. The real flow quantity between nodes is presented when the edge is selected. For example, be selecting the edge between machine 1 and loading dock 1, the flow quantity of 11 transactions can be seen. This is presented in the figure 63.

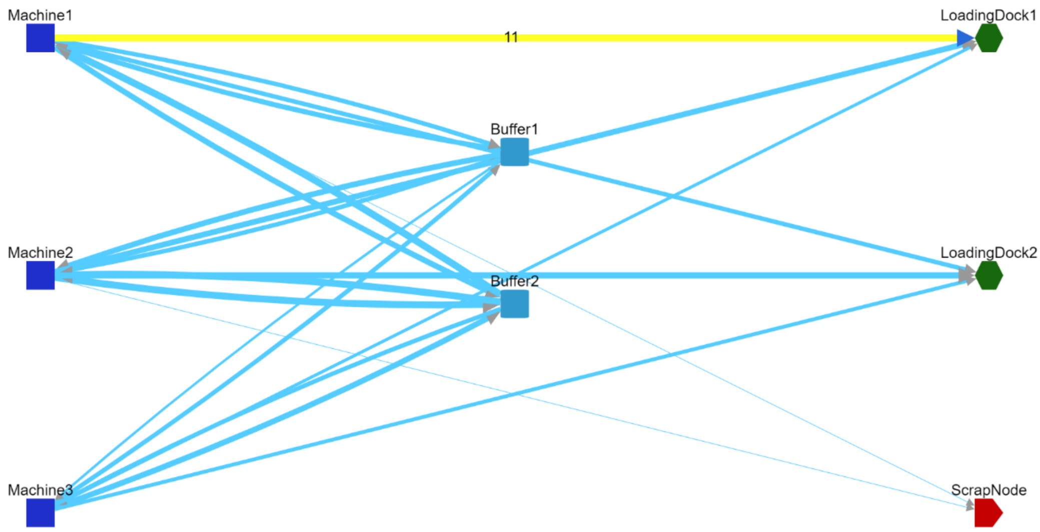


Figure 63. *By selecting the edge between two nodes, the quantity of item flow is presented*

The directed graphs of production flow can be used for designing the most effective plant layouts, to find the less used nodes and to analyse the realization of wanted route of the items. This will help to improve production flow and production planning.

4.8.1 Evaluation of the Architecture of the Designed Flow Monitoring System

Oner et al. (2016) presents suitable architecture framework for automatic identification tracking system. To design the Inspector sustainable with scientific methods, the design must be started from the base of the application. Oner et al (2016) presents 5 layers and these can be seen from the figure 40. Starting from the bottom, physical layer is similar in Inspector and in the architecture framework. Data capturing front end is also following the same design principles with RFID or barcode readers. Data capturing layer of the framework is also designed to the Inspector as a layer which collects, process and stores data to the database. There is small diversion to the Oner et al.'s (2016) framework where the database and identification is located at the next, processing modules, layer while in Inspector it is located already at the Data capturing layer.

In the Inspector, the processing modules layer is presented in this thesis. It combines the identification data to the product data and creates the reports and visualizations In the

Inspector, the application layer is the own software. With the Inspector UI, operators and other users can access and query the tracking data.

Overall, there is little differences with the example framework. The data capturing layer has larger functionality in Inspector than in framework of Oner et al. (2016). However, the architecture of the Inspector follows the example framework quite well with small modifications. Therefore, there is no need for redesign the Inspector architecture.

When designed visualization architecture of Inspector flow monitoring system is compared to the InfoVis architecture, presented in the figure 41, many similarities can be found. The first module of InfoVis is collecting, transforming and filtering the data. This is part of the layer Data collecting, storing and providing layer in Inspector. This layer transforms the collected data and stores it to the structured database for later access. Data analysis layer of Inspector is then querying the data and creates focused data sets for visualization. This is part of filtering module in InfoVis. After the data sets are created, they are mapped for visualization in the InfoVis module named mapping. In Inspector, this mapping is selecting colours, sizes and other styles for visualization. Then the data is rendered with HTML5 and JavaScript libraries. Finally, the UI of Inspector allows user interactivity to filter or sort the data or to make new data queries. This is very important to allow user to find and explore the needed data.

Inspector is following the main principles of InfoVis. Of course, using the InfoVis as base for Inspector gives many options for implementation and design. Inspector will provide multiple different kind of graphs and the target is to help the customers to understand the production, get the needed production information and gain added-value by monitoring the manufacturing flow.

5. CONCLUSIONS AND FUTURE IDEAS

Key performance indicators are widely researched, designed and used also in the manufacturing industry. KPIs have ability to combine manufacturing processes to economic values and offer vital information about production with simple numbers. That makes KPIs important for decision making and continuous improvement of production. Therefore, it is very important to understand the KPIs from many aspects. Recognizing KPI types, use cases, relations and possibilities offers create value for the companies. Detecting also the possible challenges and difficulties with KPIs, deeper understanding for KPI development and decision making can be exploited. This thesis explores usage, development and standards of KPIs for better utilization of them.

Traceability is linked to the item tracking and production flow monitoring. By tracking the item movements with simple node-based data, traceability can be achieved. Every node visit of the item leaves mark by sending an event which is stored to the database. With consecutive events, history of item flow can be achieved. Also, current location of the item is known. Inspector will utilize traceability data and visualize the data with interactive directed graphs which allows following the item movements visually.

5.1 Conclusion of Decisions for Flow Monitoring System

The target of the thesis includes finding the justification for implementing the manufacture flow monitoring system. If there are already better tools for monitoring, implementing new one is not reasonable. Therefore, research on market situation is done. There are some applications targeting to the item tracking, but these are not fitting directly to the designed market area of Inspector. Therefore, it is decided that implementing the own monitoring system could be done. Possible architecture of Inspector flow monitoring system is designed.

The KPIs for the monitoring system are developed, keeping mind that the designed node-based data is very simple. When the KPIs are formed, they are presented with ISO 22400 KPI definitions and KPI-ML definition schemas. The target is to use KPI definitions to communicate used KPIs of the flow monitoring system to possible customers. Interface to the data collecting and storing layers is designed so that KPI creation and analyses is reliable and fast.

For visualization of the KPIs, the research focuses to find possible ready visualization platform available in the market. Microsoft Power BI sounded valid option and it is tested and compared to own implementation made with Chart.js library. For the realistic data generation, simulator is built. Simulator implements designed software interfaces so that it can be utilized in the real implementation. Also, visualization server is developed with the same tools than the real implementation. After visualizations are compared, Power BI is rejected because it does not offer enough benefits to justify the high price. The features which affected to the decisions are presented in the table 28. The most relevant features are listed on the top of the table.

Table 28. *Comparing own solution with Microsoft Power BI*

Feature	Own implementation with Chart.js	Microsoft Power BI
Price	Developing costs	High costs, lot of hours for learning, integration and development
Visualization clearness	Good	Good
Visualization interactivity by default	Yes	Yes
Ease of use	Easy (if JavaScript and HTML5 is familiar)	Hard
Flexibility for data format	High	Low, data must be quite structured
Customization possibilities	High	High
Available help	High	High
Brand	Low	High

Some of the possible KPI visualizations are presented with Chart.js for justifying the idea, benefits and possibilities of flow monitoring system. Cytoscape.js library is used for traceability implementation and examples of directed graphs have been done based on realistic simulator data.

Last, the architecture of Inspector flow monitoring system is estimated in front of traceability system architecture and InfoVis architecture presented in the literature. Because the architecture of Inspector seems to fit quite well proposed architectures, no more changes for it must be done.

5.2 Future Ideas

This thesis gives good overview to KPIs and traceability, two of the important aspects in the manufacturing now and even more in the future. This is good base for further development. By storing production data and creating good interface to original Inspector, Inspector can be designed to be traceability system. By creating interfaces to other applications, Inspector could be extensive monitoring system providing KPI data about production flows, machine states and maybe even operator actions.

For tracking the full life cycle of the items, it would be interesting to monitor the whole supply chains of the customer company. It would provide value for the customer and create visibility for the supply chain. Good supply chain management can offer for example better agility for processes, smaller inventories and therefore better viability. Inspector could focus to the supply chain traceability integration without too much radical changes because it is already working as a cloud service. Moreover, the node-based flow monitoring is flexible and allows adding nodes easily, and the location of the nodes are not restricted.

One of the interesting functions for future would be concentrating on ERP interfaces. Experience from the industry shows that there is willingness in the industry to unite multiple applications, like the ERP and monitoring systems. Inspector can collect and form lots of production data and KPIs and reporting them directly to the customer ERP could be possibility. Then, the KPI-ML presented in the chapter 2.1.8 should be utilized for interface and transactions. It is well defined and offers possibility to multiple interfaces for one application. This allows using standardized message schemas and helps defining the interfaces with the customer. By using interface with standardized transaction messages, the monitoring system can also have good commercial advantage. The architecture of Inspector with KPI-ML interface is presented in the figure 64.

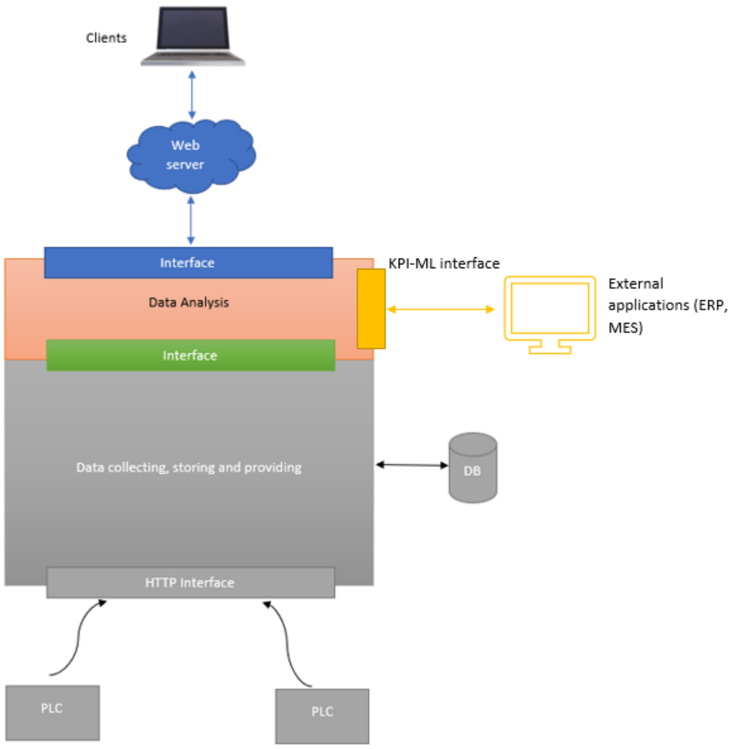


Figure 64. *Possible architecture of Inspector with KPI-ML interface to customer applications*

The KPI-ML interface would be located to Data Analysis layer because there the KPIs are created and analyzed. KPI-ML interface would allow multiple external application connections at the same time quite easily.

Soon, new node type should be added. If there are different node types for raw material buffers and WIP buffers, more information about WIP production could be offered. Then, the real WIP amount can be shown as well as the real raw material amount. In the current version of Inspector flow monitoring system, the items in every buffer node are handled as WIP which might not be sustainable with every customer. Also, adding new KPIs could be done in the future. The current data offers possibilities for KPIs like inventory turnover, rework percent, and availability rate of items which means how big percent of item types, at least one item of type, are in the storage. Of course, this needs good item type handling which is possible with the original Inspector.

Quality reporting utilizing statistical process control (SPC) methods could be implemented in the future. Then, when distractions or big variations in the KPI values are detected, these can be reported to the operators by showing or sending alarms and notifications. Implementing of alarm reporting system can be based on manually added value limits, for example for throughput time of some item type. Then, if the limits are exceeded, the alarm is sent automatically. Other, more exclusive version could calculate the value limits based on the production data. For example, if the throughput time of most of the items is between 50 and 60 minutes, the limits can be added somewhere near these values using methods of six sigma. This means that automatic reporting can be started after enough production data is collected.

One of the interesting future scopes could be adopting a route thinking to the Inspector. If the user could mark designed routes between nodes for every item type, could Inspector follow the realized route against the planned route. This would allow also controlling the production system. Then, also future node visits could be estimated and scheduled. However, this would be so major change from the monitoring system to controlling system that it should be considered very deeply. Also, Inspector as monitoring system can be installed to running system quite easily but in case of controlling, adopting Inspector to manufacturing system would be more complex.

Layout design could also be implemented when flow monitoring data is available. Maybe automatic layout design tool could be implemented in the future. Of course, there are physical limitations, but device and buffer locations based on each other could be calculated with existing data.

REFERENCES

(Bennett et al. 1999) M. Bennett, P. James, L. Klinkers, Sustainable Measures: Evaluating and Reporting of Environmental and Social Performance, Taylor and Francis, Sheffield, 1999

(BSI 2018) Balanced Scorecard Institute, Balanced Scorecard Basics, Web Site, 2018, Available (Referred 11.1.2018): http://www.balancedscorecard.org/BSC-Basics/About-the-Balanced-Scorecard

(CA 2015) CA Service Provider Center of Excellence Team, Key Performance Indicators: Establishing the Metrics that Guide Success, White Paper, 2015, 12 p Available: https://www.ca.com/content/dam/ca/us/files/white-paper/key-performance-indicators.pdf

(Cagle & Duckett 2004) K. Cagle, J. Duckett, Professional XML Schemas, WROX, 2004, 690 p.

(Caplan 1989) F. Caplan, The quality system, Chilton Book Company, Pennsylvania, 1989, pp. 45-46

(Chart.js 2018) Chart.js, Web Site, Available (Referred 27.12.2018): https://www.chartjs.org/

(Chongwatpol & Sharda 2012) J. Chongwatpol, R. Sharda, RFID-enabled track and traceability in job-shop scheduling environment, European Journal of Operational Research, 2012, published 2013, 11 p.

(Cytoscape.js 2018) Cytoscape.js, Web Site, Available (Referred 27.12.2018): http://js.cytoscape.org/#introduction/about

(Dai et al. 2014) H. Dai, M. Tseng, P. H. Zipkin, Design of traceability systems for product recall, International Journal of Production Research, Vol. 53, No. 2, 2015 (2014 online), 21 p.

(De Carlo et al. 2013) F. De Carlo, M. A. Arleo, O. Borgia, M. Tucci, Layout Design for a Low Capacity Manufacturing Line: A Case Study, Regular Paper, International Journal of Engineering Business Management, Special Issue Innovations in Fashion Industry, 2013, 10 p.

(del-Ray-Chamorro et al. 2003) F. M. del-Ray-Chamorro, R. Roy, B. van Wegen, A. Steele, A framework to create key performance indicators for knowledge managements solutions, Journal of Knowledge Management, 2003, 17 p.

(Effendi et al. 2014) M. Effendi, M.M. Razali, A. Rohana, S. Adi, A Study on The Development of Key Performance Indicators (KPIs) at an Aerospace Manufacturing Company, Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka, Industrial Campus, Melaka, Malaysia, 2014, 18 p.

(EK 2017) Confederation of Finnish Industries (EK). Available (Referred 29.12.2017): https://ek.fi/mita-teemme/talous/perustietoja-suomen-taloudesta/3998-2/

(Ezrachi 2017) A. Ezrachi, How to plan capacity for Power BI Embedded, Microsoft, Whitepaper, 2017, 6 p.

(Falcone et al. 2014) D. Falcone, A. Silvestri, V. Duraccio, V. D. Bona, Material Flow Map and Layout Optimization: action plan and software validation, Conference: XV Summer School "Francesco Turco" Impianti Industriali Meccanici, At Porto Giardino,Monopoli, 2014, 6 p.

(Fitz-Gibbon 1990) Carol Taylor Fitz-Gibbon, Performance Indicators, Bera Dialogues Number 2, Multilingual Matter Ltd, Clevedon, Philadelphia, 1990, pp. 1-4

(Fraser 2006) J. Fraser, Metrics that Matter: Uncovering KPIs that Justify Operational Improvements, Industry Directions – MESA International Research, 2006, 43 p., Available: http://www.mesa.org/files/metricspublicreport2006.pdf

(Gašpar & Mabic 2016) D. Gašpar, M. Mabic, NoSQL Databases for Big Data Management, University of Mostar, Faculty of Organization and Informatics, Croatia, 2016, 8 p.

(Hepworth 2017) K. Hepworth, Big Data Visualization: Promises & Pitfalls, Communication Design Quarterly Review, volume 4, 2017, 13p.

(Herrmann et al. 1995) J. W. Herrmann, G. Ioannou, I. Minis, R. Nagi, J. M. Proth, Design of Material Flow Networks in Manufacturing Facilities, Journal o['Manufacturing Systems, Vol 14, 1995, 13p.

(ISO 22400:2:2014)	ISO 22400:2, Automation systems and integration - Key performance indicators (KPIs) for manufacturing operations management – Part 2: Definitions and descriptions, International Organization for Standardization, First edition, 15.01.2014, 71 p.
(ISO 9000:2005)	ISO 9000:2000, Quality management systems - Fundamentals and vocabulary, International organization for Standardization, 2005
(Johnsson & Kirsch 2014)	C. Johnsson, A. Kirsch, Introducing the ISO 22400 standard, White Paper, 2014, 3 p.
(Jovan & Zorzut 2006)	V. Jovan, S. Zorzut, Use of Key Performance Indicators in Production Management, Department of Systems and Control Jožef Stefan Institute, Ljubljana, Slovenia, 2006, 6 p.
(Jovane et al. 2009)	F. Jovane, E. Westkämper, D. Williams, The ManuFuture Road: Towards Competitive and Sustainable High-Adding-Value Manufacturing, Edition illustrate, Springer, 2009, 12 p.
(Kang et al. 2016)	N. Kang, C. Zhao, J. Li, J. A. Horst, A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems, International Journal of Production Research, 2016, 19 p.
(Kaushik 2010)	A. Kaushik, Web Analytics 2.0. The Art of Online Accountability & Science of Customer Centricity, Wiley Publishing, Inc., 2010, p. 37.
(Keeple et al. 2003)	J. Keeple, S. Topiol, S. Berkeley, Using Indicators to Measure Sustainability Performance at a Corporate and Project Level, Journal of Business Ethics, Arthur D. Little Limited, 2003, 10 p.
(Khabbazi et al. 2010)	M. R. Khabbazi, M. D. Yusof Ismail, N. Ismail, S. A. Mousavi, Modeling of Traceability Information System for Material Flow Control Data, Australian Journal of Basic and Applied Sciences, INSInet Publication, 2010, 9 p.
(Kvarnström 2008)	B. Kvarnström, Traceability Methods for Continuous Processes, Licentiate thesis, Luleå University of Technology, 2008, 49 p.

(Lindberg et al. 2015)	C-F. Lindberg, S. Tan, J. Yan, F. Starfelt, Key performance indicators improve industrial performance, The 7th International Conference on Applied Energy – ICAE2015, Elsevier Ltd., 2015, 6 p.
(Liu et al. 2014)	S. Liu, W. Cui, Y. Wu, M. Liu, A survey on information visualization: recent advances and challenges, Springer-Verlag Berlin Heidelberg, 2014, 21 p.
(Lukkari 2018)	P. Lukkari, How to know that IoT has landed on your shop floor?, Fastems Blog, 2018, Available (Referred 13.08.2018): https://blog.fastems.com/how-to-know-that-iot-has-landed-on-your-shop-floor/
(MedBiquitous 2004)	MedBiquitous Technical Steering Committee, XML Schema Design Guidelines, Version 1.3, 2004, 18 p.
(MESA 2015)	MESA International, Key Performance Indicator Markup Language, Version 1, May 2015, 45 p.
(Microsoft docs 2018)	Microsoft docs, web-site, Available (Referred 7.11.2018): https://docs.microsoft.com/en-us/
(Mulrane 2016)	P. Mulrane, Why is it important to monitor performance in production lines?, FlexLink, Available: http://blog.flexlink.com/why-is-it-important-to-monitor-performance-in-production-lines/
(Müller & Oehm 2018)	R. Müller, L. Oehm, Process industries versus discrete processing: how system characteristics affect operator tasks, Springer-Verlag London Ltd, April 2018, 20 p.
(Nair & Shah 2007)	B. Nair, M. Shah, Compliance and Traceability in Manufacturing, Aberdeen Group, Boston, 2007, p. 8
(Ofori-Boateng 2017)	C. Ofori-Boateng, The History Of KPIs And Their Rise To Popular Use Today, Christiansteven Software Business Intelligence Blog, Available (Referred 11.1.2018): https://go.christiansteven.com/the-history-of-kpis-and-their-rise-to-popular-use-today
(Oner et al. 2016)	M. Oner, A. Ustundag, A. Budak, An RFID-based tracking system for denim production processes, Int J Adv Manuf Technol, 90:591–604, 2016, 14 p.

(PingFlow 2018)	PingFlow, Website, Available (Referred 1.11.2018): https://www.pingflow.com/
(Plex 2018)	Plex, Website, Available (Referred 22.10.2018): https://www.plex.com/
(Power BI 2018)	Microsoft Power BI, Website, Available (Referred 5.11.2018): https://powerbi.microsoft.com/en-us/
(Rakar et al. 2004)	A. Rakar, S. Zorzut, V. Jovan, Assesment of Production Performance by Means of KPI, Control 2004, University of Bath, UK, 2004, 5 p.
(Rakar & Zorzut 2010)	A. Rakar, S. Zorzut, Key Performance Indicators for Production Management, Department of System and Control, Jožef Stefan Institute, Ljubljana, Slovenia, 2010, 2 p.
(Scoreboard 2018)	Scoreboard, KPI Examples, Web-site, Available (Referred 4.1.2018): https://kpidashboards.com/kpi/
(Staniszewski et al. 2014)	M. Staniszewski, S. Legutko, P. Raos, Production Data Collection, Exposure and Analysis is a Small Production Enterprise, Tehnički vjesnik 21, 2014, 6 p.
(Steele 1995)	D. C. A. Steele, Structure for lot-tracing design, Production and Inventory Management Journal, First Quarter, 1995, pp. 53-59
(Töyrylä 1999)	I. Töyrylä, Realising the potential of traceability –A case study research on usage and impacts of product traceability, Acta Polytechnica Scandinavia, Mathematics, Computing and Management, Espoo, 1999, 216 p.
(van Dorp 2002)	K.-J. van Dorp, Tracking and tracing: a structure for development and contemporary practices, Logistics Information Management, Volume 15, Number 1, 2002, 10 p.
(van Eekelen 2008)	J.A.W.M. van Eekelen, Modelling and control of discrete event manufacturing flow lines, Eindhoven University of Technology, 2008, 254 p.
(W3Schools 2018)	W3Schools, XML Schema Tutorial, Available (Referred 17.3.2018): https://www.w3schools.com/xml/schema_intro.asp

(Wang 2014) K.-S. Wang, Intelligent and integrated RFID (II-RFID) system for improving traceability in manufacturing, Shanghai University and Springer-Verlag Berlin Heidelberg, 2014, 15 p.

(Weber & Thomas 2005) A. Weber, R. Thomas, Key Performance Inducators: Measuring and Managing the Maintenance Function, Ivara Corporation, Burlington Ontario, Canada, 2005, 15 p.

(White et al. 2007) G. R. T White, G. Gardiner, G. Prabhakar, A. A. Razak, A Comparison of Barcoding and RFID Technologies in Practice. Journal of Information, Information Technology and Organizations. 2. 10.28945/142, 2007, 14 p.

(Zebra 2018) Zebra, Web Site, Available (Referred 31.10.2018): https://www.zebra.com/

(Zhu et al. 2018) L. Zhu, C. Johnsson, M. Varisco, M. M. Schiraldi, Key performance indicators for manufacturing operations management – gap analysis between process industrial needs and ISO 22400 standard, 8th Swedish Production Symposium, SPS 2018, Sweden, May 2018, 7 p.

APPENDIX A: KPI DEFINITION SCHEMA FOR SELECTED KPIS AND EXAMPLE XML-FILES

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" at-
tributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="GuidType">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-
      F]{4}-[0-9a-fA-F]{12}"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="RangeType">
    <xs:all>
      <xs:element name="ID" type="xs:string" />
      <xs:element name="Description" type="xs:string" />
      <xs:element name="LowerLimit" type="xs:string" />
      <xs:element name="UpperLimit" type="xs:string" />
    </xs:all>
  </xs:complexType>

  <xs:complexType name="KPIDefinitionType">
    <xs:sequence>
      <xs:element name="ID" type="GuidType" />
      <xs:element name="Description" type="xs:string" />
      <xs:element name="Name" type="xs:string" />
      <xs:element maxOccurs="unbounded" name="Scope" type="xs:string" />
      <xs:element name="Formula" type="xs:string" />
      <xs:element name="UnitOfMeasure" type="xs:string" />
      <xs:element name="Range" type="RangeType"/>
      <xs:element name="Trend" type="xs:string" />
      <xs:element maxOccurs="unbounded" name="Timing" type="xs:string" />
      <xs:element maxOccurs="unbounded" name="Audience" type="xs:string" />
      <xs:element maxOccurs="unbounded" name="ProductionMethodology"
      type="xs:string" />
      <xs:element name="Notes" type="xs:string" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="KPIDefinition" type="KPIDefinitionType"/>
</xs:schema>
```

```xml
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>63f3833f-5a2b-47e1-b0e1-9d508c7d6136</ID>
    <Description>
      The average inventory for items in a time frame. Calculated as weighted
      arithmetic mean. w is weight in seconds and x is value in pieces.
    </Description>
    <Name>Average Inventory</Name>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula>Average Inventory = (w1*x1+w2*x2+…+wn*xn)/(w1+w2+…+wn)</Formula>
    <UnitOfMeasure>pcs</UnitOfMeasure>
    <Range>
        <ID>NaturalUnlimited</ID>
        <Description>Natural Unlimited Range</Description>
        <LowerLimit>0</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend></Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Beginning and ending time can be selected. Also, some specific WIP buffer
      can be selected instead of all buffers. A specific item can be selected.
    </Notes>
</KPIDefinition>
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>2c058d9e-9bbd-4bc0-8760-7131f4edca7b</ID>
    <Description>
      The scrap quantity (SQ) compared to the total produced quantity (TPQ)
    </Description>
    <Name>Scrap Ratio</Name>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula>Scrap Ratio = (SQ/TPQ)*100</Formula>
    <UnitOfMeasure>%</UnitOfMeasure>
    <Range>
        <ID>Natural</ID>
        <Description>Natural Range</Description>
        <LowerLimit>0</LowerLimit>
        <UpperLimit>100</UpperLimit>
    </Range>
    <Trend>Smaller-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Timing>Real-Time</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Beginning and ending time can be selected. Some specific items can be se-
      lected. Scrap after some specific node can be selected.
    </Notes>
</KPIDefinition>
```

```xml
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">

    <ID>3b38a1c7-0bc3-44d0-9d17-8e9725b49358</ID>
    <Description>
      Time required for item to pass through a manufacturing process. Throughput
      time is difference between output node time (ONT) and first node time (FNT)
    </Description>
    <Name>Throughput Time</Name>
    <Scope>Item</Scope>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula>Throughput Time = ONT - FNT</Formula>
    <UnitOfMeasure>TimeSpan</UnitOfMeasure>
    <Range>
        <ID>TimeSpan</ID>
        <Description>TimeSpan Range Unlimited</Description>
        <LowerLimit>0s</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend>Smaller-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Some specific item can be selected
      Average time is provided for multiple items or an item types and exact time
      for specific item.
    </Notes>
</KPIDefinition>
```

```xml
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>1e29c6d0-10a7-4c59-95a4-2fa844b52c6b</ID>
    <Description>
      Time required for a raw material conversion to a ready product. Time spent
      in value-adding nodes (NnT).
    </Description>
    <Name>Process Time</Name>
    <Scope>Item</Scope>
    <Scope>Item types</Scope>
    <Scope>All item types</Scope>
    <Formula>Process Time = N1T + N2T +..+ NnT</Formula>
    <UnitOfMeasure>TimeSpan</UnitOfMeasure>
    <Range>
        <ID>TimeSpan</ID>
        <Description>TimeSpan Range Unlimited</Description>
        <LowerLimit>0s</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend>Smaller-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Some specific item can be selected
      Average time or for multiple items
      Process time for single node can be selected
    </Notes>
</KPIDefinition>
```

```xml
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>712f7064-2e64-48cb-8aaf-cf2891e3c36b</ID>
    <Description>
      Time required from moving items between nodes. (MnTn)
    </Description>
    <Name>Move Time</Name>
    <Scope>Item</Scope>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula>Move Time = M1T + M2T +..+ MnT</Formula>
    <UnitOfMeasure>TimeSpan</UnitOfMeasure>
    <Range>
        <ID>TimeSpan</ID>
        <Description>TimeSpan Range Unlimited</Description>
        <LowerLimit>0s</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend>Smaller-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Total moving time for a specific item or average moving time for an item
      type.
      Average move time for all items.
      Average time between two nodes for single or all items.
    </Notes>
</KPIDefinition>
```

```xml
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>c26eaa25-32de-4749-ae3c-9fb90999dba8</ID>
    <Description>
      Time spent in buffers and storages. (BnTn)
    </Description>
    <Name>Queue Time</Name>
    <Scope>Item</Scope>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula>Queue Time = B1T + B2T +…+ BnT</Formula>
    <UnitOfMeasure>TimeSpan</UnitOfMeasure>
    <Range>
        <ID>TimeSpan</ID>
        <Description>TimeSpan Range Unlimited</Description>
        <LowerLimit>0s</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend>Smaller-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Total queuing time for specific item or average queuing time for item type.
      Average queuing time for all items.
    </Notes>
</KPIDefinition>
<KPIDefinition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:sche-
maLocation="http://Inspector.fi/KPIDefinition.xsd">
    <ID>4be701e3-8cf1-498c-920d-5ee8eaf8c521</ID>
    <Description>
      Number of ready items in a time frame.
    </Description>
    <Name>Output Rate</Name>
    <Scope>Item type</Scope>
    <Scope>All item types</Scope>
    <Formula></Formula>
    <UnitOfMeasure>pcs</UnitOfMeasure>
    <Range>
        <ID>NaturalUnlimited</ID>
        <Description>Natural Unlimited</Description>
        <LowerLimit>0</LowerLimit>
        <UpperLimit>Unlimited</UpperLimit>
    </Range>
    <Trend>Higher-is-better</Trend>
    <Timing>On-demand</Timing>
    <Timing>Periodically</Timing>
    <Audience>Operator</Audience>
    <Audience>Supervisor</Audience>
    <Audience>Management</Audience>
    <ProductionMethodology>Discrete</ProductionMethodology>
    <ProductionMethodology>Batch</ProductionMethodology>
    <Notes>
      Can be calculated for specific item type or for multiple item types. Time
      frame can be changed.
    </Notes>
</KPIDefinition>
```

APPENDIX B: IFLOWANALYTICSCORE INTERFACE FUNCTIONS

GetAverageInventory			
Used to get average inventory size at given time period. Using different parameters, including only specific buffers or item type is possible			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
type	string	yes (“”)	Wanted item type. Default value for all item types.
buffers	List<Guid>	yes (null)	Wanted buffers. Default value for all buffers.
resolution	TimeResolution	yes (0)	Wanted time resolution. Default value for all.
Return value	**Return value data type**	**Return value default**	
	Dictionary<Guid, List<float>>	new Dictionary<Guid, List<float>>()	Average inventory size for item types (dictionary) in different times based on resolution (list)

GetScrapPercentage			
Used to get scrap percentage at given time period. Using different parameters, including only specific item type is possible. Using parameters, the scrap percentage after specific nodes can be found			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
type	string	yes (“”)	Wanted item type. Default value for all part types.
nodes	List<Guid>	yes (null)	Scrap after specified nodes. Default value for all scrap.
Return value	**Return value data type**	**Return value default**	
	float	0	Scrap percentage for wanted parameters.

GetThroughputTimeForSingleUnit			
Used to get throughput time for single item. Using parameters, throughput time between specific nodes can be found			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
serialNumber	string	no	Item serial number
startNode	Guid	yes (new Guid())	Wanted start node
endNode	Guid	yes (new Guid())	Wanted end node
Return value	**Return value data type**	**Return value default**	
	long	0	Throughput time

GetThroughputTimeForMultipleItems			
Used to get throughput time for multiple item types in given time frame. Using parameters, throughput time between specific nodes can be found			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
itemTypes	List<string>	no	Types of wanted items
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
startNode	Guid	yes (new Guid())	Wanted start node
endNode	Guid	yes (new Guid())	Wanted end node
Return value	**Return value data type**	**Return value default**	
	Diction-ary<Guid, List<long>>	new Dictionary<Guid, List<long>>	Throughput time for wanted item types

GetProcessTimeSingleUnit			
Used to get process time for single item.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
serialNumber	string	no	Item serial number
Return value	**Return value data type**	**Return value default**	
	Diction-ary<Guid, long>	new Dictionary<Guid, long>()	Process time for item in every value-adding node

GetProcessTimeForMultipleItems			
Used to get process time for multiple item types in given time frame			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
types	List<string>	no	Types of wanted items
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
Return value	**Return value data type**	**Return value default**	
	Dictionary<Guid, Dictionary<Guid, long>>	new Dictionary<Guid, Dictionary<Guid, long>>()	Process time for wanted item types in every value-adding node

GetBufferTimeSingleItem			
Used to get buffer/queuing time for single item.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
serialNumber	string	no	Item serial number
Return value	**Return value data type**	**Return value default**	
	long	0	Time spend in buffers for item

GetBufferTimeMultipleItems			
Used to get buffer/queuing time for multiple item types.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
types	List<string>	no	Types of wanted items
Return value	**Return value data type**	**Return value default**	
	long	0	Average time spend in buffers for item type

GetOutputRate			
Used to get output rate. Using parameters, time resolution of output can be selected, also specific item type can be selected.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
resolution	TimeResolution	yes (0)	Wanted time resolution
type	string	yes (“”)	Wanted item type
Return value	**Return value data type**	**Return value default**	
	Dictionary<Guid, List<int>>	new Dictionary<Guid, List<int>>()	Output rate for item types (dictionary) in different time resolution (list)

APPENDIX C: IFLOWROUTESCORE INTERFACE FUNCTIONS

GetRouteForItem			
Used to get route for specific item from node to node.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
serialNumber	string	no	Serial number of a specific item
Return value	**Return value data type**	**Return value default**	
	List<Tuple<Guid, DateTime>>	new List< Tuple<Guid, DateTime>>()	List of node visits for item. Tuple Item1 is node identifier and Item2 is visit time

GetRoutesForItemType			
Used to get routes for items of specific type during wanted time range.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
itemType	string	no	Wanted item type
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
Return value	**Return value data type**	**Return value default**	
	List<List<Tu-ple<Guid, DateTime>>>	new List<List<Tu-ple<Guid, DateTime>>>()	List of lists of item node visits. Tuple Item1 is node identifier and Item2 is visit time. First list contains items and second list visits of spe-cific items.

GetRoutes			
Used to get routes for items during wanted time range.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
startTime	DateTime	no	Wanted start time of time period
endTime	DateTime	no	Wanted end time of time period
Return value	**Return value data type**	**Return value default**	
	List<List<Tuple<Guid, DateTime>>>	new List<List<Tuple<Guid, DateTime>>>()	List of lists of item node visits. Tuple Item1 is node identifier and Item2 is visit time. First list contains items and second list visits of specific item.

GetNodeItems			
Used to get items in a node in specific time.			
Parameter name	**Parameter data type**	**Parameter optionality (default value)**	**Parameter description**
nodeId	Guid	no	Identifier of the node
time	DateTime	yes (DateTime.Now)	Wanted time. If empty, current time is used.
Return value	**Return value data type**	**Return value default**	
	List<string>	new List<string>()	List of item serial numbers in a node.